UNIVERSIDAD CARLOS III DE MADRID
Dpto. de INGENIERÍA DE SISTEMAS Y AUTOMÁTICA

TRABAJO FIN DE GRADO

DISEÑO DEL ROBOT HUMANOIDE TEO PARA SU SIMULACIÓN EN V-REP

GRADO EN INGENIERÍA ELECTRÓNICAINDUSTRIAL Y AUTOMÁTICA

Autor: Eduardo Vázquez Muñoz
Profesor: Santiago Martínez de la casa Díaz

Fecha: 18/09/2014

Resumen

La ciencia está investigando y desarrollando robots humanoides con fines tales como la interacción con herramientas y entornos humanos, con fines experimentales, como el estudio de la locomoción bípeda, o para otros fines. En la actualidad los robots con desplazamiento sobre ruedas son más eficaces que los robots con desplazamiento bípedo.

Desde el departamento de ingeniería de sistemas y automática están investigando y creando un robot humanoide real del modelo TEO para simular entornos humanos mediante simuladores y luego poder implementarlos en el robot real.

Con este trabajo fin de grado lo que se intenta conseguir es la creación tridimensional del robot TEO para importar en el simulador V-REP, con ello, se permitirá la simulación de tareas del robot humanoide TEO en un entorno virtual, con lo que podremos observar las características necesarias para su movimiento.

Índice

Contenido

Índice de figuras:

1 Introducción

1.1 Introducción

Entendemos comúnmente por robot: “Máquina electrónica que puede ejecutar automáticamente distintas operaciones o movimientos”.

Un robot es por tanto una máquina capaz de realizar tareas repetitivas de forma más barata, rápida y precisa que los propios seres humanos, interactuando además con su entorno. Es por ello, por lo que podríamos definir la robótica humanoide como el área de la ingeniería que se encarga de desarrollar sistemas robotizados que pretenden imitar las condiciones del ser humano.

Precisamente con el fin de imitar al ser humano, el aspecto físico del robot será lo más parecido posible al humano, por lo que dispondrá de dos piernas que le proporcionaran movimiento, dos brazos con los que se pueda equilibrar o sujetarse y finalmente de una cabeza que, si bien no es esencial, le proporcionara un aspecto estéticamente más humano.

Se puede decir que los robots humanoides poseen una ventaja sobre otro tipo de robots, siendo esta la de poder trabajar directamente en el mismo entorno que los propios humanos con la peculiaridad de que dicho entorno no ha de ser modificado. Sin embargo y como consecuencia de esta ventaja la complejidad en el diseño y control aumentan sustancialmente.

Dicha complejidad que viene aparejada con los robots humanoides hacen que tanto su diseño como construcción tengan un elevado coste. Es por esto, por lo que se debe proporcionar a los usuarios un simulador de robots que les facilite un entorno seguro en que puedan desarrollar las pruebas que necesiten sin riesgo de dañar a los autómatas.

En este proyecto vamos a utilizar la herramienta de Solidworks para el diseño del robot humanoide TEO y el simulador V-REP para importar el modelo resultante del robot humanoide TEO y proceder así a la realización de simulaciones.

1.2 Motivación

Actualmente se dan muchas áreas de investigación en torno a la robótica humanoide si bien, aun, no han salido al mercado robots fiables y robustos de tamaño humano. Además la bipedestación aun no se encuentra resuelta completamente.

En nuestra universidad el departamento de ingeniería de sistemas y automática se encarga de investigar y construir un robot humanoide real del modelo TEO. Para ello, se necesita crear el diseño real de TEO mediante Solidworks con las medidas exactas del robot real, una vez construido el modelo, en el simulador V-REP se simularán algunos tipos de movimiento para comprobar su correcto funcionamiento.

El fin de estas investigaciones es llegar a construir robots bípedos que realicen tareas versátiles y que puedan desplazarse por terrenos irregulares ya que en la actualidad los robots con desplazamiento mediante ruedas son bastante más eficientes y proporcionan mayor velocidad a las tareas programadas.

Con todo esto lo que se pretende es el acercamiento a la realización más precisa de tareas propias del ser humano así como una mayor interactuación con ellos.

Con este programa lo que se trata es de permitir al usuario crear y modificar un robot manipulador que imita el comportamiento del ser humano. Con la simulación lo que conseguiremos será poder evaluar las posibles utilidades del robot y con ello realizar infinidad de pruebas que nos permitan un ahorro tanto en costes como en recursos y tiempo.

Tratamos por tanto de conseguir que este proyecto sirva de plataforma para probar los movimientos de un robot humanoide.

1.3 Objetivos

Este proyecto surge con la idea de crear el modelo del Robot Humanoide TEO con la herramienta de diseño Solidworks para poder importar el modelo al simulador V-REP a partir de un modelo tridimensional existente. Con este modelado se podrá simular las tareas del robot humanoide TEO en un entorno virtual, permitiendo así comprobar las características necesarias para el movimiento del mismo.

En primer lugar, se debe realizar un estudio exhaustivo del programa de Solidworks y del simulador V-REP así como el análisis de sus características principales con el objetivo de familiarizarse con el entorno de trabajo para poder crear el robot siguiendo todas las características cinemáticas del robot humanoide TEO.

Cuando tengamos el sólido de cada eslabón diseñado en Solidworks, tenemos que conseguir importar todos los sólidos y ensamblarlos mediante articulaciones en el simulador V-Rep, una vez tengamos el robot construido para su manejo en el simulador, desarrollaremos mediante cinemática inversa y cinemática directa algunos movimientos con el robot.

2 Robot Humanoide

2.1 Historia de la robótica

La palabra "robot", es de origen checo y significa “siervo o esclavo”; fue acuñada por el escritor checo Karel Capek (1890-1938) en su obra teatral R.U.R., estrenada en Europa en 1920. [1]

Con la creación de maquinas obedientes y autómatas, se conseguiría uno de los sueños del ser humano, que esas maquinas pudiesen llevar a cabo los trabajos duros y repetitivos para los que no es necesaria ninguna capacidad de improvisación.

Los comienzos de la robótica actual pueden datarse en la industria textil del siglo XVII, cuando se creó una maquina textil programable mediante tarjetas perforadas. El inventor fue Joseph Jacquard en 1801. Posteriormente, con la llegada de la Revolución Industrial se potenciaron este tipo de creaciones.

Una de las primeras muñecas mecánicas que se conocen fue construida por Henri Maillardert en 1805, la cual, podía realizar dibujos gracias a una serie de levas que hacían la función de “programa” a través del cual podía escribir y dibujar.

Con la creación de sistemas de cadenas de montajes que pueden dividir la fabricación de cualquier objeto en tareas pequeñas y con la llegada de la Revolución Industrial, se dio un primer gran paso para la robotización total en las industrias

Popularmente se define al concepto de robot como “aquel elemento de apariencia humana que actúa como tal”. Este concepto es el que la ciencia ficción ha usado y desarrollado en multitud de ocasiones.

La lógica evolución de la robótica nos lleva a que se estén desarrollando cada vez robots más inteligentes y con más y mejores extensiones sensoriales con lo que poder entender sus acciones y captar el mundo que los rodea. Esto lo que permitirá es que lleguen a tomar decisiones propias y a controlar incluso el proceso en tiempo real.

2.2 Conceptos básicos

La definición adoptada por el Instituto Norteamericano de Robótica aceptada internacionalmente para Robot es: "Manipulador multifuncional y reprogramable, diseñado para mover materiales, piezas, herramientas o dispositivos especiales, mediante movimientos programados y variables que permiten llevar a cabo diversas tareas."[1]

Tal y como venimos apuntando podemos definir Robot como un dispositivo mecánico creado para realizar tareas automáticas y repetitivas. Estas tareas deben ser programadas por el ser humano.

Los robots suelen ser reprogramables y multifuncionales, con conexión de retroalimentación y cuya inteligencia viene dada por una computadora o un micro controlador que ejecuta un programa. Sin embargo se han desarrollado mucho los Robots con inteligencia alámbrica, cuyas acciones son generalmente llevadas a cabo por máquinas que mueven extremidades o impulsan al robot.

El autor Isaac Asimov en ciencia ficción propuso tres leyes fundamentales de la robótica. Estas leyes son unas normas y comportamientos que deberían respetar siempre los robots para cuando convivieran con los seres humanos y fueran independientes. Las tres leyes representan el código moral del robot.

> **Primera ley:** Un robot no puede hacer daño a un ser humano o, por inacción, permitir que un ser humano sufra daño. [1]
>
> **Segunda ley:** Un robot debe obedecer las órdenes dadas por los seres humanos, excepto si estas órdenes entrasen en conflicto con la 1ª Ley. [1]
>
> **Tercera ley:** Un robot debe proteger su propia existencia en la medida en que esta protección no entre en conflicto con la 1ª o la 2ª Ley. [1]

Esta redacción de las leyes es la forma convencional en la que los humanos de las historias las enuncian; su forma real sería la de una serie de instrucciones equivalentes y mucho más complejas en el cerebro del robot.

2.3 Actualidad en robots humanoides

Actualmente los robots están siendo desarrollados con el objetivo de conseguir una apariencia, en la forma física y los movimientos del ser humano, ofreciendo sensaciones de un propósito propio. La independencia de sus movimientos hace que sus acciones sean la razón de un estudio razonable.

Hoy en día los robots más utilizados en investigaciones robóticas han sido los robots manipuladores, los móviles y los robots con más de 2 patas.

Un diseño humanoide puede tener fines funcionales, tales como la interacción con herramientas y entornos humanos, con fines experimentales, como el estudio de la locomoción bípeda, o para otros fines. En general, los robots humanoides tienen un torso, una cabeza, dos brazos y dos piernas, aunque algunas formas de robots humanoides pueden modelar sólo una parte del cuerpo, por ejemplo, de la cintura para arriba. Algunos robots humanoides pueden tener cabezas diseñadas para replicar los rasgos faciales humanos, tales como los ojos y la boca.

La finalidad de los robots humanoides frente a los robots industriales, es el entorno de trabajo. Los humanoides podrán trabajar en un entorno humano como viviendas, lugares públicos o privados, etc.

Respecto al estado actual de las investigaciones en robots humanoides he de destacar que:

- Se sigue desarrollando la locomoción bípeda y estable, ya que la complejidad de estos movimientos no está controlados completamente.

- La complejidad de desarrollo de un robot humanoide con funciones similares a las del ser humano son tan complicadas de igualar a las funciones del ser humano que los robots que se fabrican son de tamaños compactos para abaratar costes y facilitar más el control y sus operaciones.

A continuación, se presenta los mejores desarrollos en robots humanoides: [2]

2.3.1 Nao Robot

Nao (pronunciado *noa*) es un robot humanoide programable y autónomo, desarrollado por la empresa francesa Aldebaran robotics, desarrolla muchísimas cualidades motrices y una gran interactividad con el entorno. Se comenzó a desarrollar en 2004 con fines de investigación y educación. En diciembre de 2011 se lanza Nao Next gen (ver figura 1) que es su última versión, tiene un software mejorado, más potente y cámaras de alta definición. Muestra los grandes avances en el desarrollo de robots humanoides, donde no solo puede hacer tareas repetitivas sino que puede realizar muchísimas tareas dependiendo el entorno y que se puede adaptar a nuestras necesidades ya que es programable. El nuevo robot no ha cambiado mucho de aspecto, pero estrena como cerebro de operaciones a un procesador **Intel Atom a 1.6GHz**, e incorpora dos cámaras de alta definición que prometen un mejor reconocimiento de objetos y rostros, incluso en situaciones de poca luz. En cuanto al reconocimiento de voz, cuenta con cuatro micrófonos, y de ello se encarga el software especializado **Nuance**, también han llegado mejoras importantes en los algoritmos que gestionan el caminar y las colisiones.

Entre sus cualidades principales se destacan su reconocimiento de voz y de órdenes, puede detectar las diferentes formas de los objetos, rostros y seguir su moviendo, es sensible al tacto en muchas partes de su cuerpo, tiene conectividad Wi-Fi que inclusive le permite comunicarse con otros robots de su misma clase, entre muchos otros. [3]

Figura 1: Nao Robot

2.3.2 New Asimo

ASIMO, es el más famoso de los robots humanoides creado por Honda, la finalidad de Honda para ASIMO era la de pretender ayudar a las personas que carecen de movilidad completa en sus cuerpos, así como para animar a la juventud para estudiar ciencias y matemáticas. Muestra grandes cualidades para la translación bípeda. Fue el que dio el gran primer paso en lograr que un robot caminara y es uno de los mejores robots de la actualidad (ver figura 2).

Figura 2: New Asimo

Este robot tiene 5 capacidades principales, y son las siguientes:

Está formado por dos cámaras que funcionan como los ojos, a los que se suman unas líneas que forman una sonrisa que lo hace más amigable a las personas, es capaz de predecir trayectorias de objetos, apartándose si algo viene hacia él, pudiendo moverse a 9 kilómetros por hora con sus piernas. Se rediseñaron sus manos para trabajar en el lenguaje de signos, dándole más accesibilidad a personas con capacidades diferentes, sumado a su mejorado reconocimiento de personas y voces. Por último, se presentaron novedades en cuanto a la interacción con objetos, como llevar una bandeja, destapar una botella, y servir su contenido en un vaso. [4]

2.3.3 HRP-4C

HRP-4C es un robot humanoide, creado por el Instituto Nacional de Tecnología y Ciencia Industrial Avanzada de Tokio, también conocido por como AIST, y presentado al público en 2009.

Por su fisonomía femenina puede ser considerada como una ginoide. Su inteligencia artificial le permite el reconocimiento del habla, contando también con la capacidad de síntesis del habla. No fue diseñado para el servicio del hombre así como lo hace ASIMO, sino que fue diseñado puramente para el entretenimiento, especialmente en el campo de la moda (ver figura 3). Las dimensiones de su cuerpo han sido tomadas de una media de mujeres japonesas, y se desplaza mediante 30 motores, siendo capaz de adoptar poses.

Su altura es de 1.58 metros y pesa 43 Kg. Incluyendo la batería. Cuenta con 42 motores en total, que sirven para que realice todos sus movimientos de elegancia y coqueteo tratando de imitar a las modelos. En su rostro cuenta con 8 motores que le permiten realizar gestos (permitiéndole mostrar varias emociones) e imitar el movimiento de los labios cuando habla.

Figura 3: HRP-4C

Posee una inteligencia artificial que le permite el reconocimiento del habla, es decir, puede percibir cuando una persona le está hablando y puede entender varias órdenes sencillas.

También cuenta con la capacidad de síntesis del habla, que en otras palabras se refiere a que su voz no es producida por grabaciones, sino que posee un software y hardware que convierte texto en sonido, lo que permite que sus diálogos sean fácilmente programable. [5]

2.3.4 Partner Robot

Figura 4: Partner Robot

Los Toyota Partner Robots o Robots acompañantes Toyota son una serie de robots humanoides desarrollados por Toyota. Debutaron tocando música en tambores y trompetas (ver figura 4) en la EXPO Mundial 2005 en Aichi, Japón. Hay 5 robots en total, la mayoría de los cuales tienen diferentes sistemas de movimiento: Versión 1 (robot bípedo), Versión 2 (con ruedas similares a un Segway), versión 3 (ruedas tipo Segway), versión 4 (sistema de cableado único) y el i-Foot (montable con 2 piernas). En julio de 2009, Toyota dio a conocer un video de las corrientes y permanentes habilidades de su robot acompañante. El robot alcanza los 7 km/h, sin embargo caminar y correr sólo lo pueden hacer en superficies planas. [6]

2.3.5 QRIO

Figura 5: QRIO

QRIO ("Quest for cuRIOsity") un robot Humanoide desarrollado por Sony, QRIO es para SONY, lo que ASIMO para HONDA, Ante la dificultad de construir un robot tamaño humano (desde 1,50 metros de altura) capaz de correr SONY apostó por un pequeño robot humanoide (ver figura 5), ligero y con unas articulaciones extremadamente fuertes, así desarrollo el robot ahora llamado QRIO. Este robot de SONY consiguió correr estableciendo el ansiado hito en Diciembre de 2003 cuando fue presentado a la prensa con un video que enseñaba al robot corriendo. Dispone de una tecnología denominada "Intelligent Servo actuador" que es lo que le permite andar dinámicamente es decir puede variar las r.p.m. y el torque en las articulaciones, y emplea una técnica denominada "Zero Moment Point" para mantener la estabilidad. [7]

3 Herramientas de diseño

3.1 Solidworks

SolidWorks es un programa de diseño asistido por computadora para modelado mecánico desarrollado para el sistema operativo Microsoft Windows. Es un modelador de sólidos paramétrico.

El programa permite modelar piezas y conjuntos y extraer de ellos tanto planos técnicos como otro tipo de información necesaria para la producción. Es un programa que funciona con base en las nuevas técnicas de modelado con sistemas CAD. El proceso consiste en traspasar la idea mental del diseñador al sistema CAD, "construyendo virtualmente" la pieza o conjunto. Posteriormente todas las extracciones (planos y ficheros de intercambio) se realizan de manera bastante automatizada. [8]

Todo el robot TEO ha sido diseñado por esta herramienta elemento a elemento y luego ensamblado en V-REP.

Cada eslabón del robot, ha sido creado desde el modelo de la pieza de cada parte del robot, para el diseño de los eslabones en Solidworks hemos aprendido a crear un croquis de diseño, como extruirlo para crear el modelo en tres dimensiones y donde esta cada herramienta en el programa.

En la pantalla principal (ver figura 6) nos encontramos con la pantalla de planos donde vamos a poder crear, visualizar y mover para su total vista los modelos, en la parte superior estará la barra de herramientas en función de la necesidad del diseño y en el lateral izquierdo un gestor de los diseños creados, planos, y pantallas de edición donde colocar las dimensiones y datos específicos de cada figura.

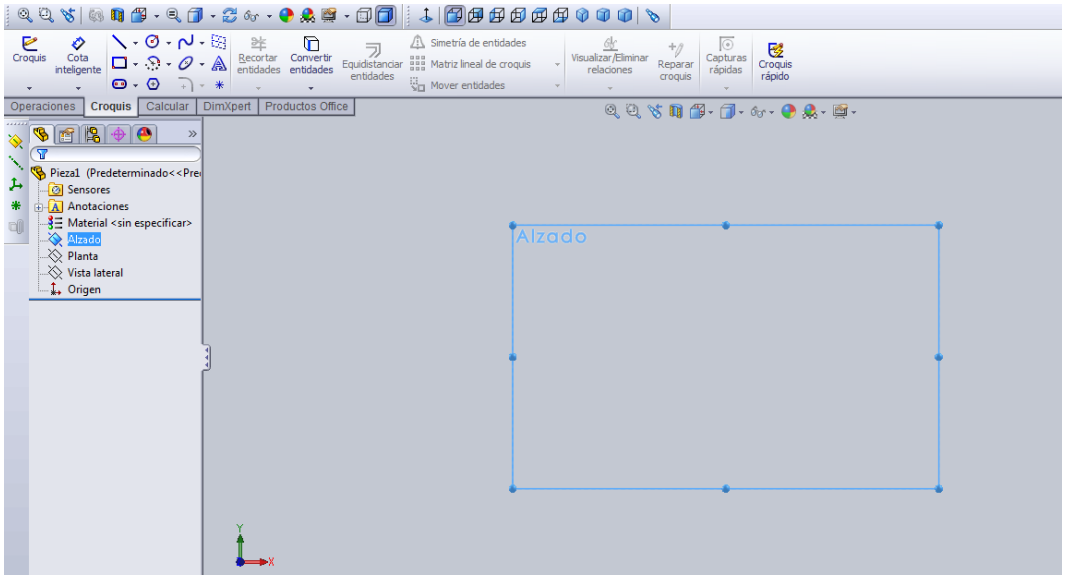

Figura 6: Interfaz Solidworks

Un ejemplo del modelo diseñado es el muslo de la pierna que mostramos a continuación (ver figura 7).

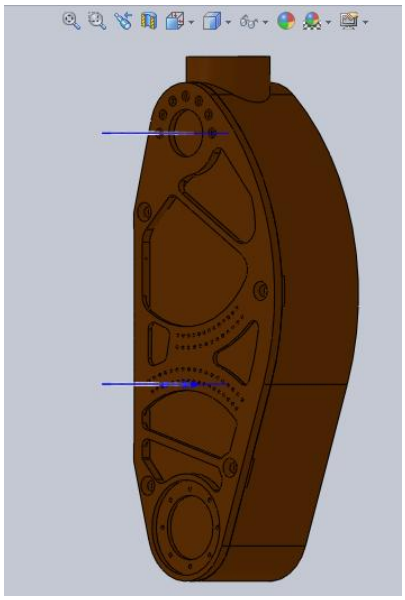

Figura 7: Muslo derecho del robot Teo

Para crear este sólido, primero tenemos que tener el diseño de la pieza, a partir de esta podemos crear sólidos en tres dimensiones para que luego se puedan ensamblar y sean equivalentes a las medidas y eslabones del robot real Teo.

Lo primero para comenzar a trabajar en Solidworks es escoger el plano de la pieza que vamos a trabajar (ver figura 8), en nuestro caso cada plano será la pieza de cada parte del robot hasta que construyamos el sólido completo, partiendo como base las medidas y diseño del robot Teo.

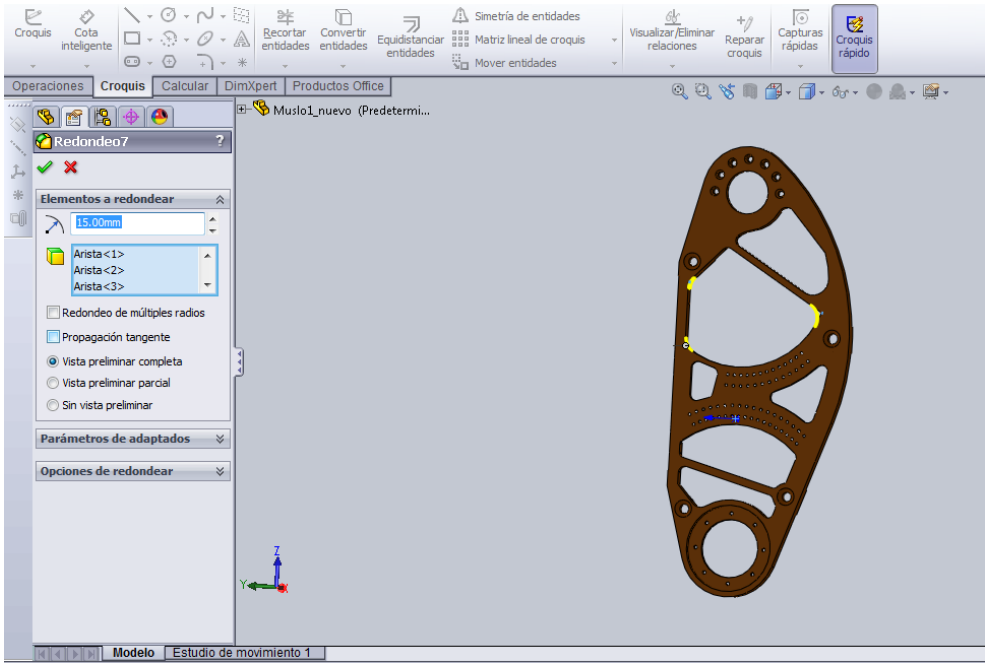

Figura 8: Plano de la pieza a diseñar

Una vez que se tiene la pieza en la pantalla principal del programa, lo siguiente es generar el sólido a partir de un croquis (ver figura 9), en la barra de herramientas podemos elegir la forma en la que queremos el croquis.

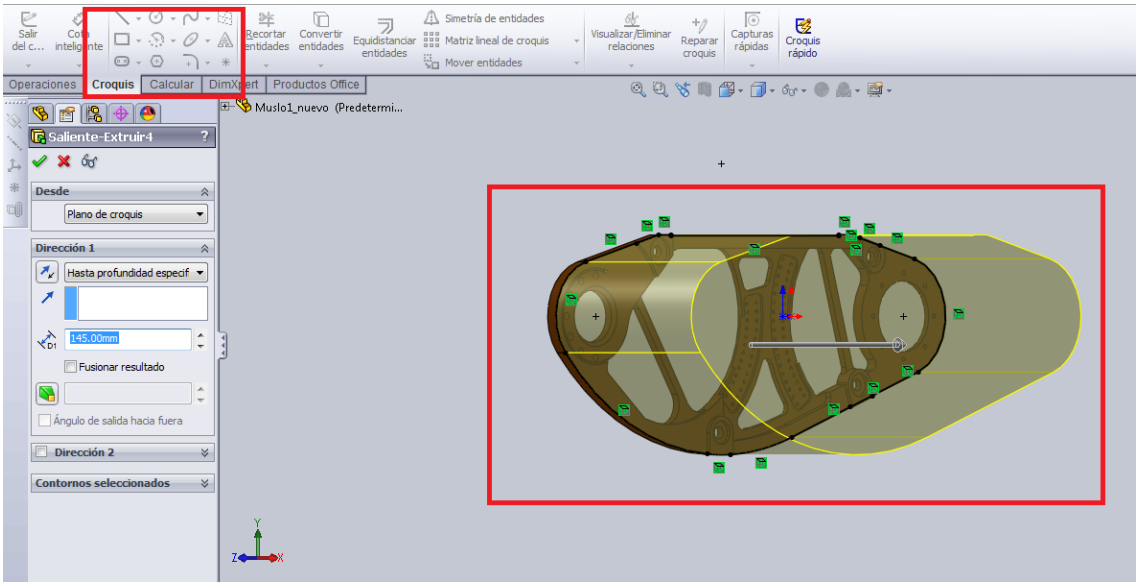

Figura 9: Solido a partir de un croquis

Con el croquis realizamos un diseño acotando siempre sus dimensiones de lo que queremos extruir, nos dirigimos a la pestaña de operaciones y hacemos clic en “extruir saliente/base” en las propiedades nos pedirá que seleccionemos el croquis anteriormente dibujado, una vez seleccionado, el programa nos muestra una vista

previa de la extrusión (ver figura 10), en la misma pestaña de propiedades podemos editar la extrusión anotando distancias a extruir, dirección de la extrusión, anotando las medidas exactas que tenemos del robot real Teo para hacerlo lo más exacto posible.

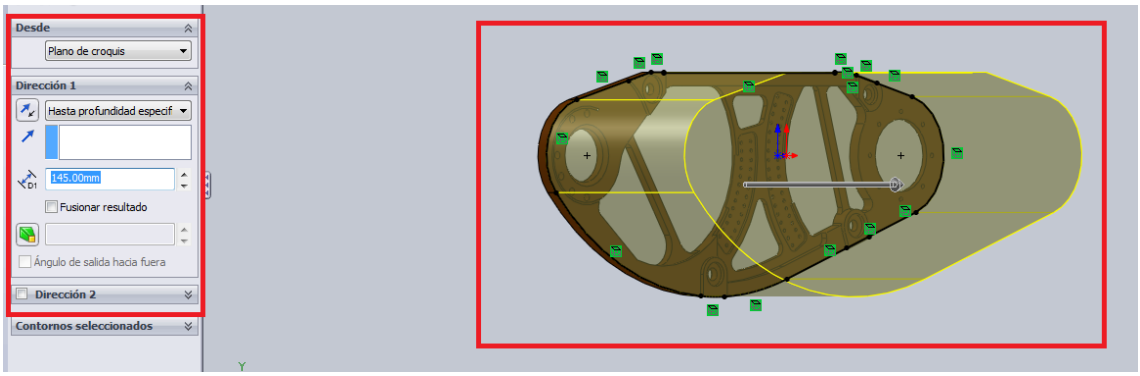

Figura 10: Ajustes de las dimensiones del solido

Una vez que tenemos las medidas de la extrusión y la dirección finalizamos cada tarea con el tic verde que aparece en la barra de propiedades (ver figura 11).

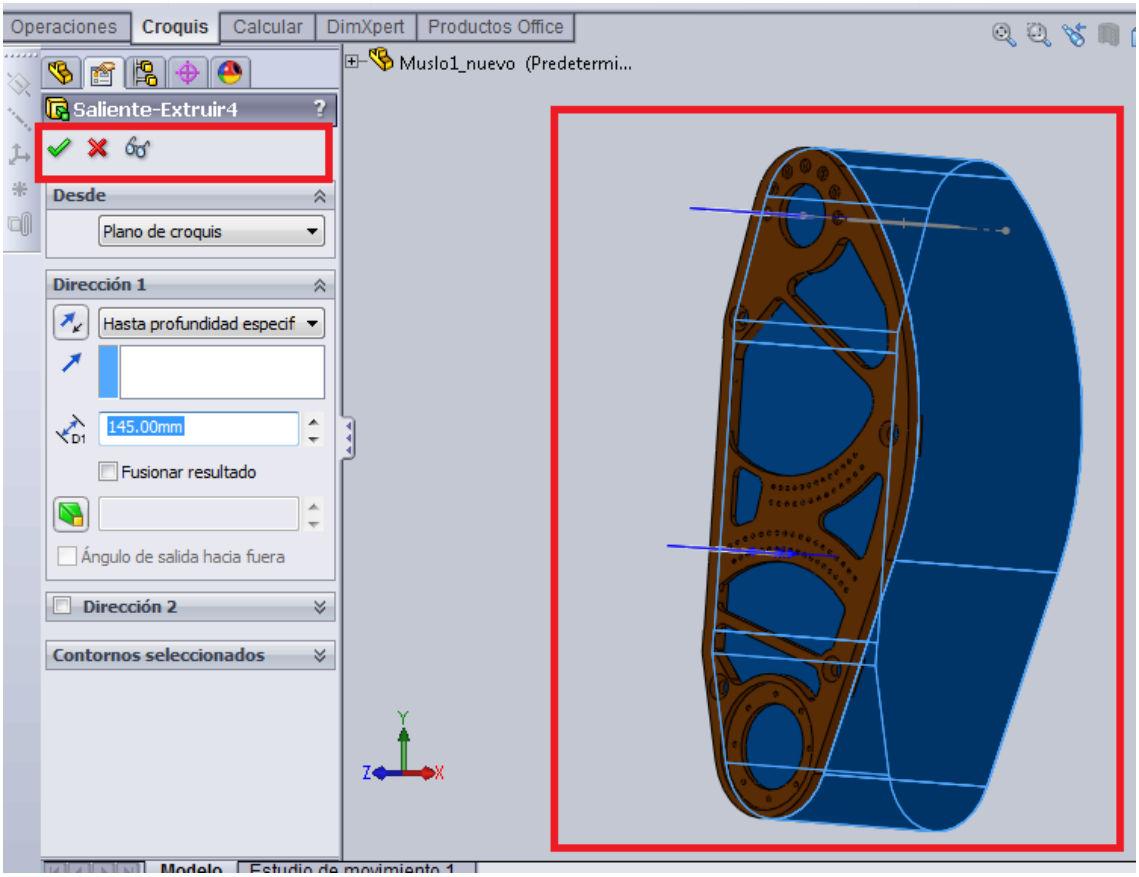

Figura 11: Medidas del solido

Cuando lo que queremos es editar la extrusión creada para juntar con otro diseño y que quede ensamblado y tenemos que extruir corte (ver figura 12).

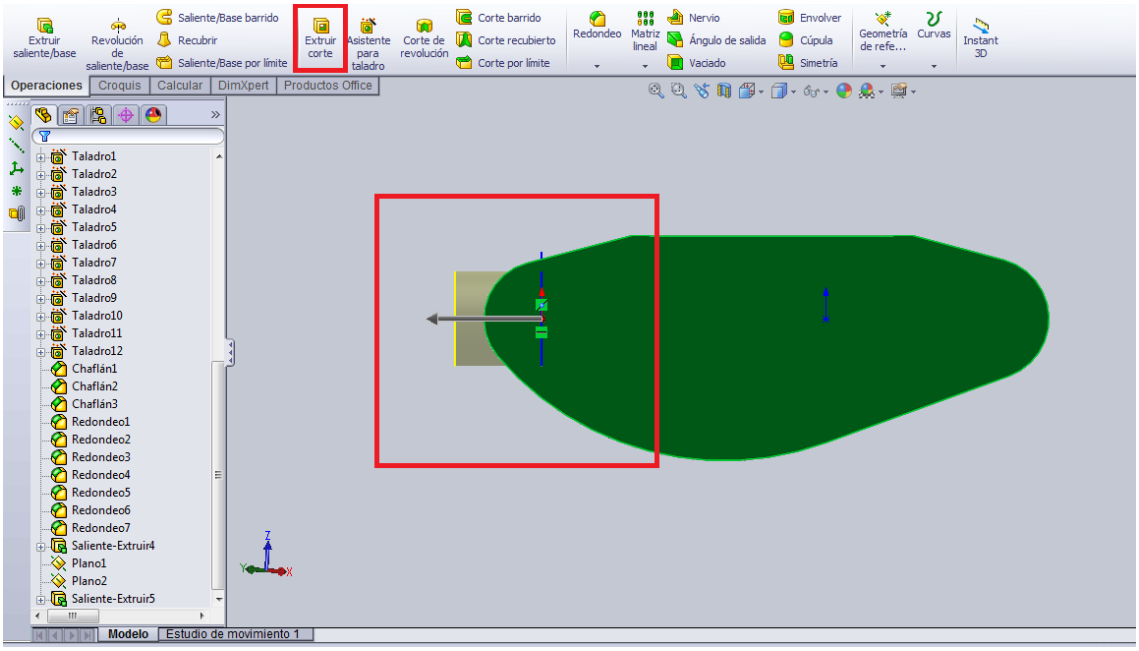

Figura 12: Ensamblado de diseño dentro del diseño creado

Con el sólido anterior lo que hacemos en crear un croquis nuevo del diseño del corte que queremos realizar a nuestro solido (ver figura 13), todos los croquis tienen que diseñarse bajo superficies planas.

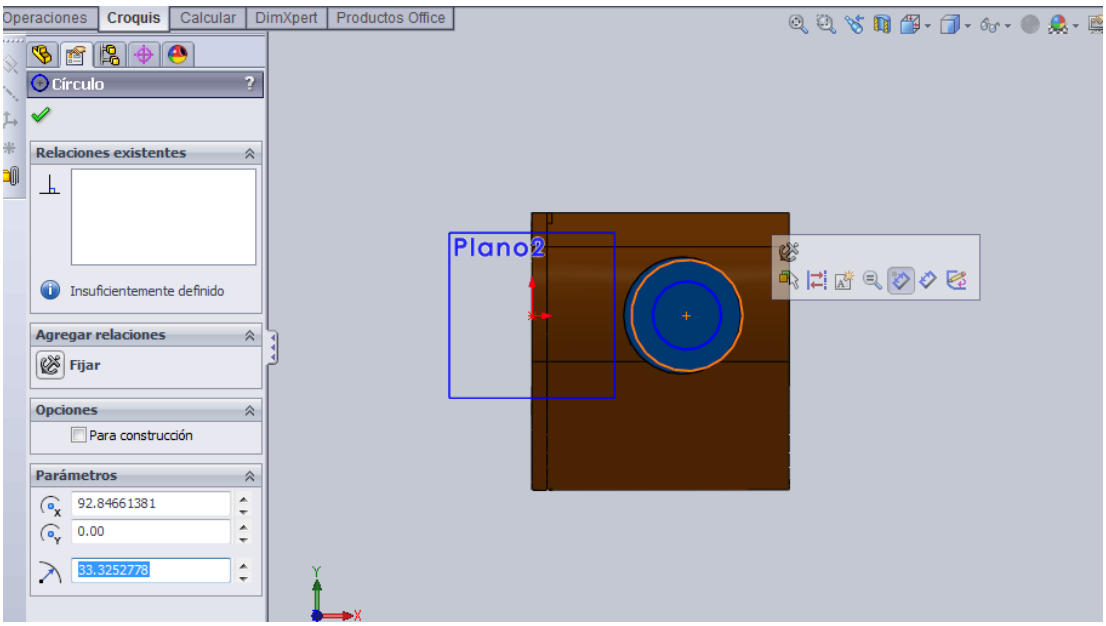

Figura 13: Croquis de los nuevos cortes en el diseño

Salimos del croquis después de haberlo dibujado y nos vamos a la opción de operaciones y seleccionamos de la barra de herramientas la opción de extruir corte y seleccionamos e perfil del croquis anteriormente diseñado que cortara a la extrusión, podemos dimensionar el corte que deseemos con las medidas que buscamos (ver figura 14).

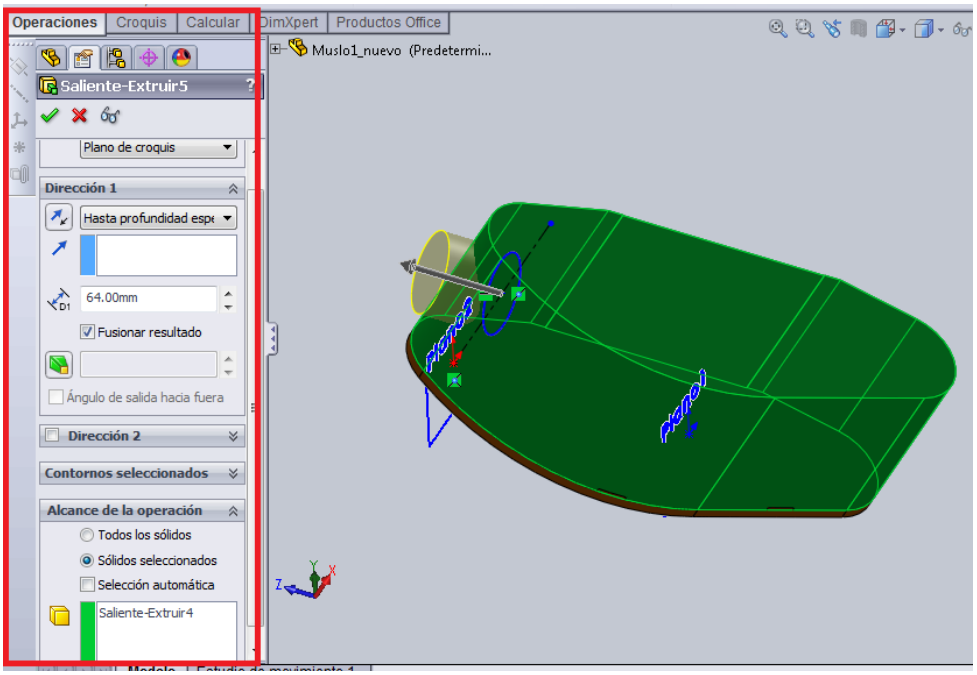

Figura 14. Dimensiones del corte

Cada pieza del robot ha sido extruida y dimensionada con las medidas del robot Teo, en las siguientes imágenes podemos ver más modelos de creación como por ejemplo la tibia, el hombro y la muñeca, desde la pieza hasta su modelo en tres dimensiones por Solidworks (ver figuras 15 a 24).

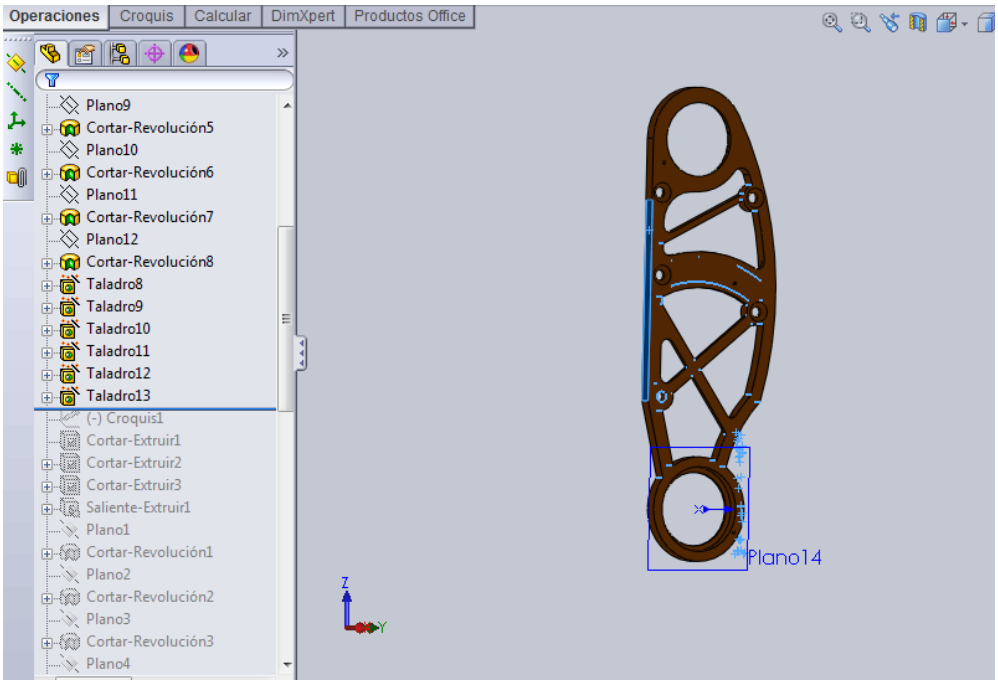

Figura 15: Modelo de la tibia de Teo

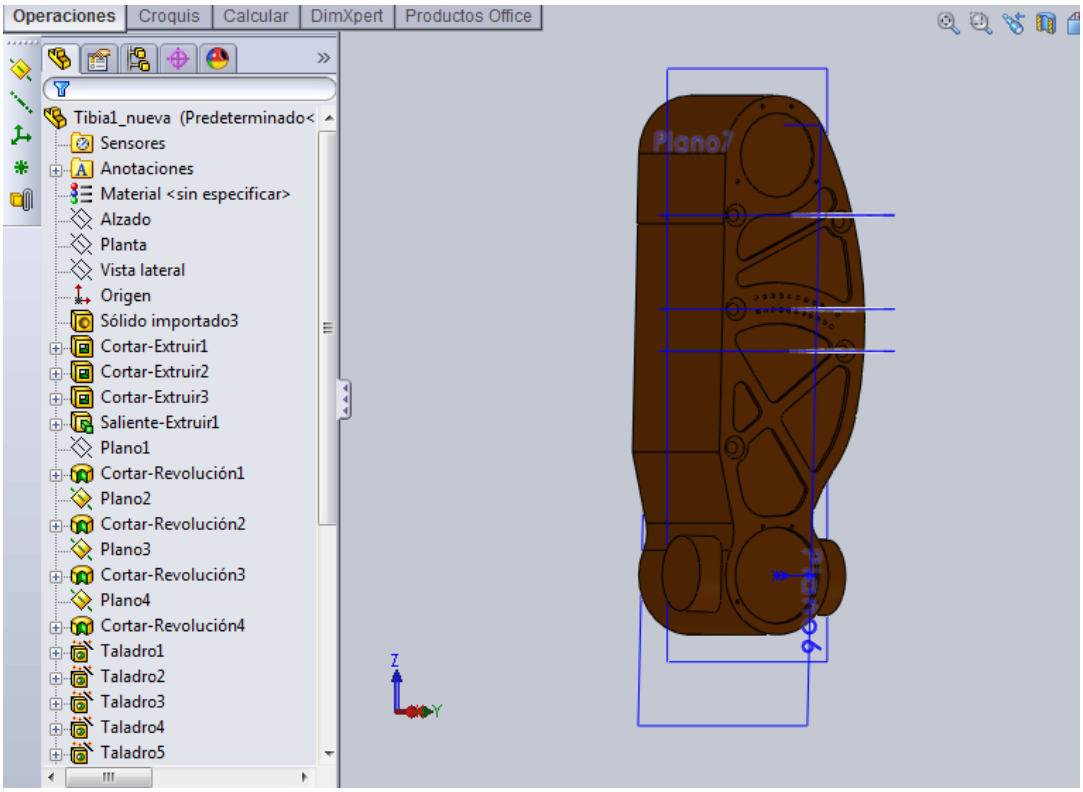

Figura 16: Solido de la tibio de Teo

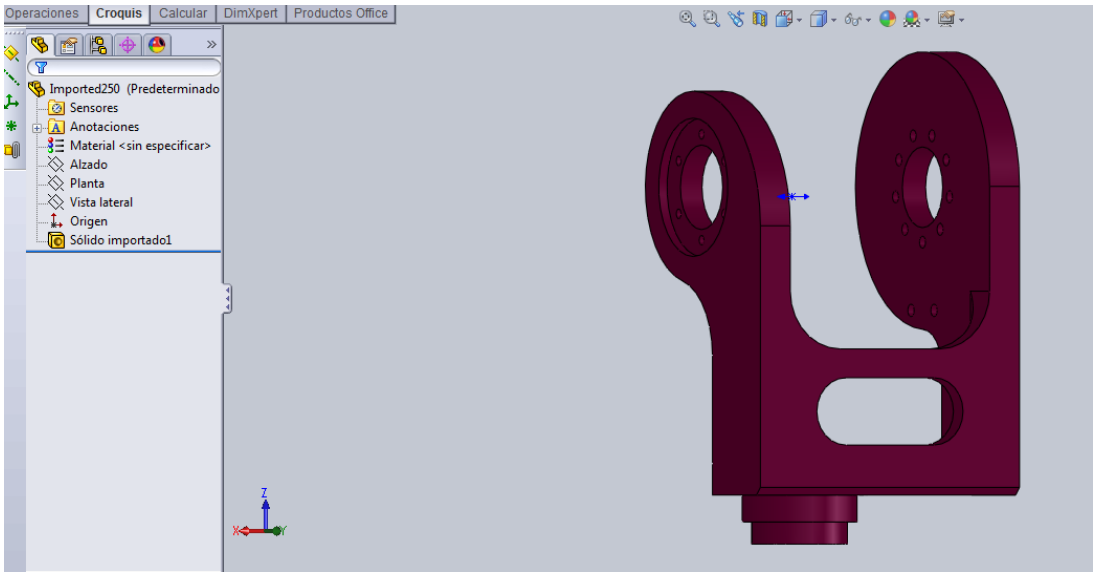

Figura 17: Modelo principal del hombro

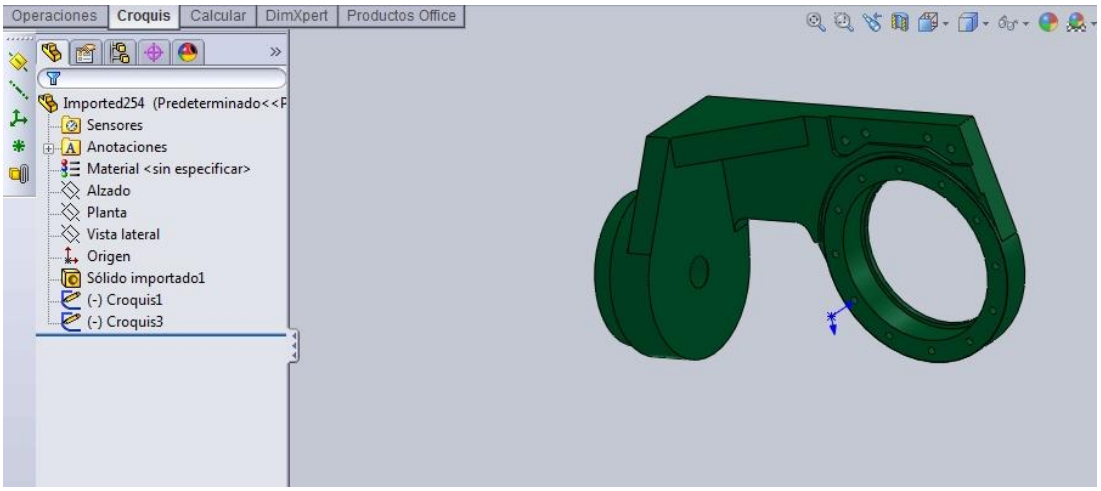

Figura 18: Parte del hombro

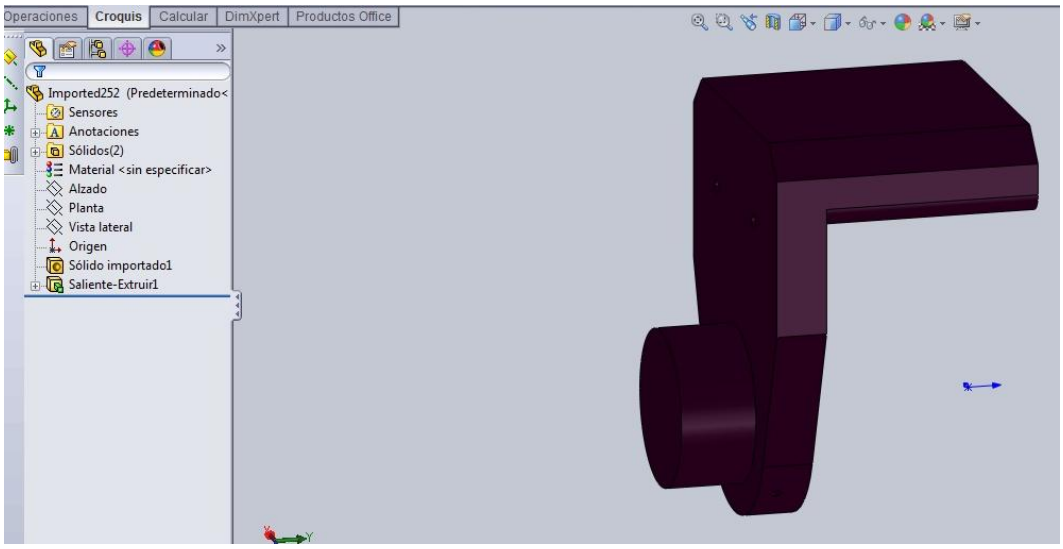

Figura 19: Parte del hombro

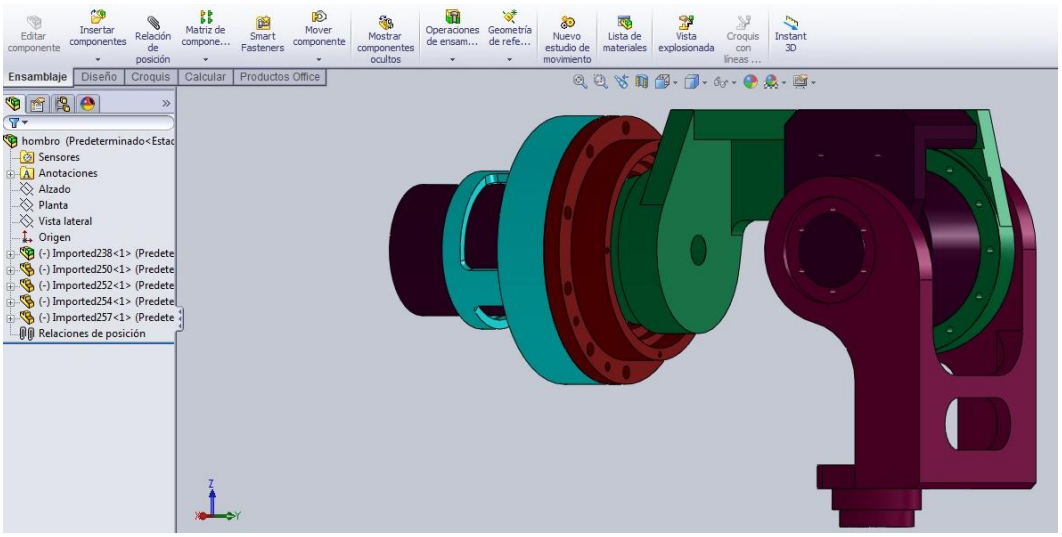

Figura 20: Solido del hombro de Teo

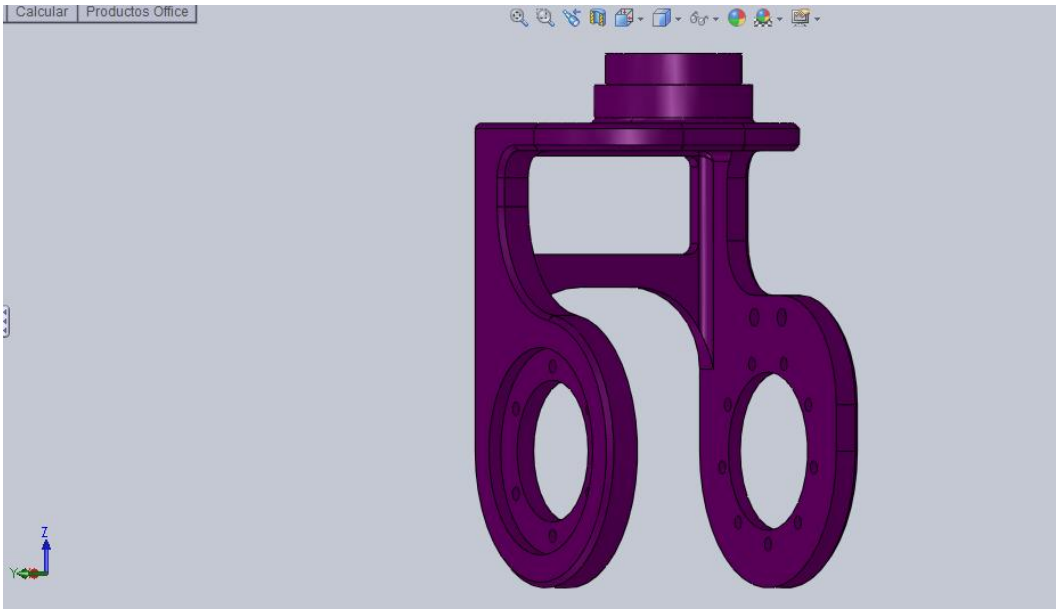

Figura 21: Modelo principal de la muñeca

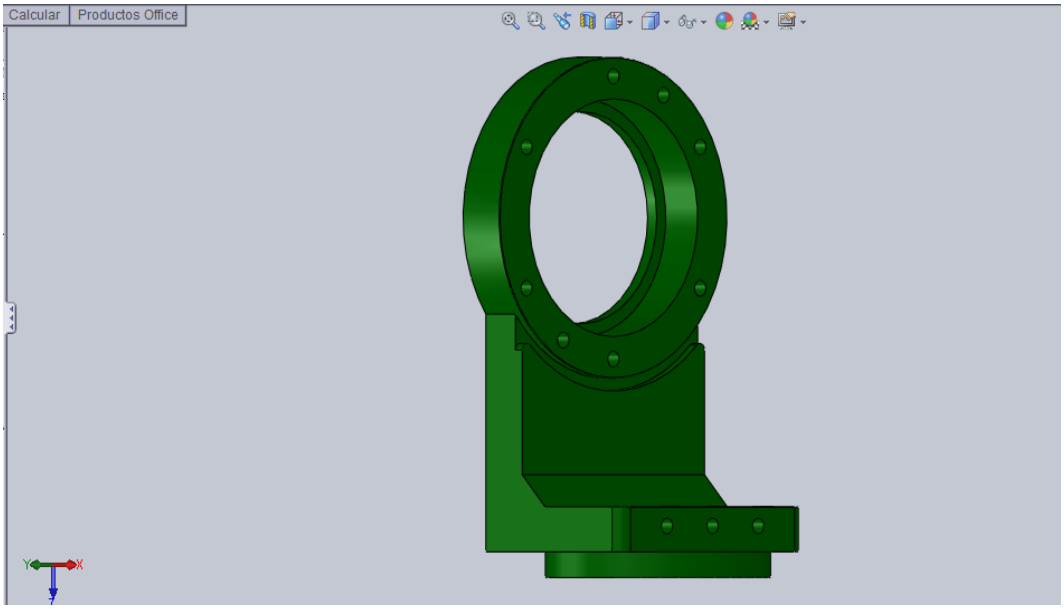

Figura 22: Parte de la muñeca

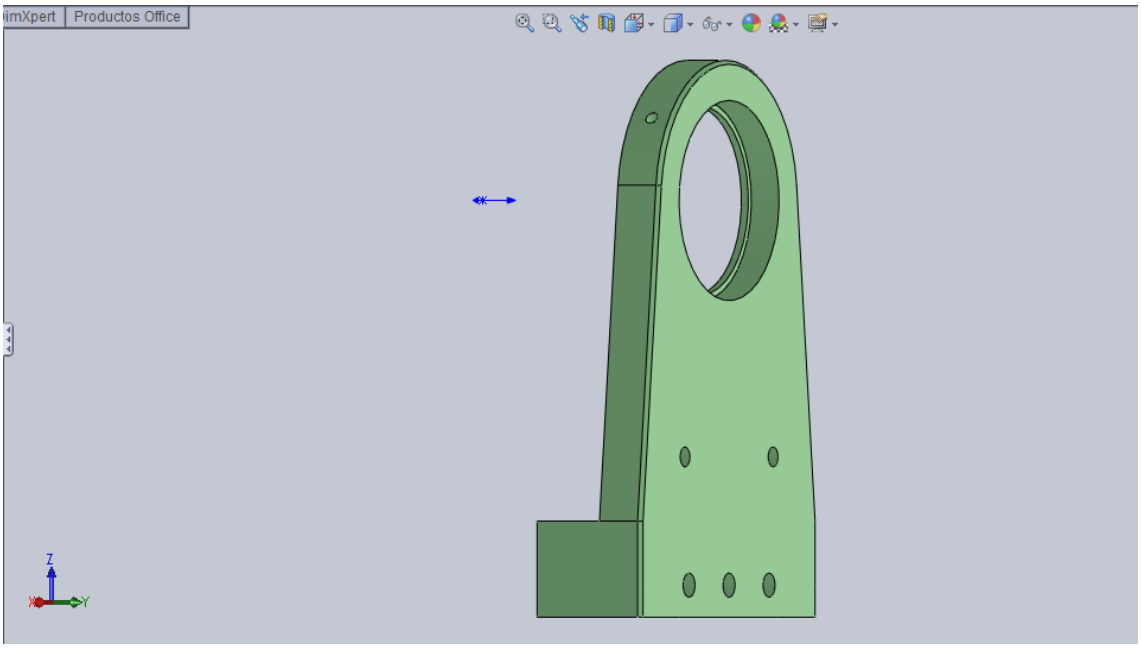

Figura 23: Parte de la muñeca

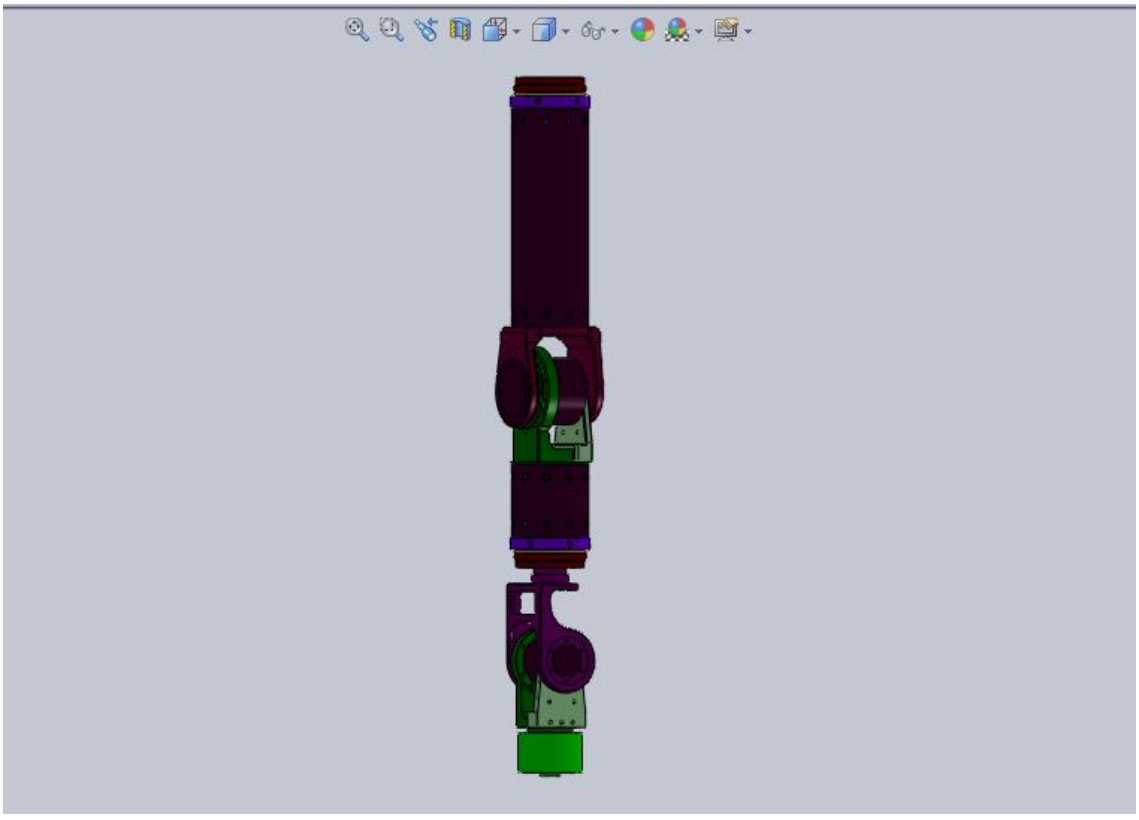

Figura 24: Solido del brazo de Teo

4 Diferentes simuladores para robótica

4.1 Introducción

Si queremos modelar y manejar un robot en entornos virtuales es necesario el uso de simuladores para poder evaluar el funcionamiento del diseño creado y poder aplicar situaciones reales en modelos virtuales evitando posibles errores en ajustes, poder testear el diseño antes de trasladarlo al entorno real y simular diferentes hipótesis de funcionamiento para terminar de consolidar y verificar que los ajustes y diseños han sido óptimos para su finalidad y así ahorrar en el coste final del diseño real.

En la actualidad existe una gran cantidad de herramientas que permiten realizar simulaciones de robots en un ambiente de tres dimensiones (3D). La interacción entre los objetos creados se realiza mediante modelado, teniendo en cuenta la física del sólido rígido.

La mayoría de simuladores utilizan interfaces gráficas para la construcción de los robots y el ambiente en el cual interactúan, y todas admiten diferentes lenguajes para la programación de los controladores, tales como PYTON, Java,C/C++, etc.

Sobre la amplia gama de simuladores disponibles se ha decidido utilizar la aplicación V-REP.

4.2 Diferentes Simuladores

A continuación comentaremos los principales simuladores que hay en el mercado.

4.2.1 SimRobot

El simulador SimRobot (ver figura 25) ha sido desarrollado en conjunto por la Universidad de Bremen y el centro de inteligencia artificial alemana con el objetivo de investigar sobre robot autónomos.

Simrobot es capaz de simular robots definidos por el usuario arbitrarias en el espacio tridimensional. Incluye un modelo físico que se basa en la dinámica de cuerpos rígidos. Para permitir una amplia flexibilidad en la construcción de modelos precisos, una variedad de diferentes cuerpos genéricos, sensores y actuadores se ha implementado. Además, el simulador sigue un enfoque orientado al usuario mediante la inclusión de varios mecanismos para la visualización, manipulación actuador directo, y la interacción con el mundo simulado. [9]

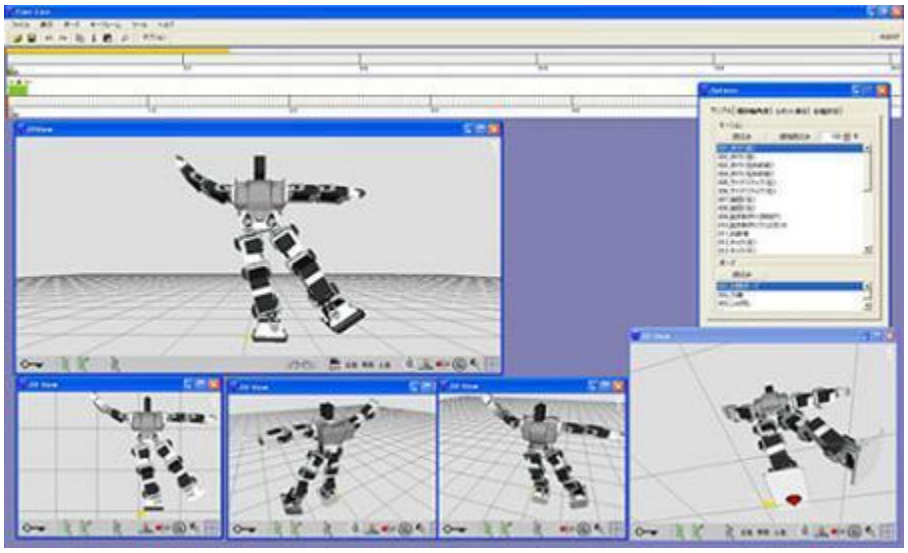

Figura 25: Simulador SimRobot

4.2.2 Gazebo

Gazebo ofrece la posibilidad de simular con precisión y eficacia las poblaciones de robots en entornos interiores y exteriores complejas. Dispone de gráficos de alta calidad, y las interfaces programáticas y gráficas convenientes. Lo mejor de todo, Gazebo es gratuito con una vibrante comunidad

Fue desarrollado por el Dr. Andrew Howard y Nate Koenig en el año 2002 en la Universidad del Sur de California. Gazebo es un simulador 3D, que permite cargar diferentes tipos de robots, sensores y actuadores dentro de un mundo artificial. Gazebo ofrece las primitivas para leer de los sensores e interactuar sobre sus actuadores. También se usa ODE y OpenGL para la simulación de cuerpos rígidos y la visualización de las imágenes.

Sus características principales son: Dispone el modelo de cámara estéreo, genera mapas estéreos de profundidad de la imagen. El interfaz gráfica de usuario escrita en wxPython y se puede usar sin necesidad de instalar el servidor Player. [10]

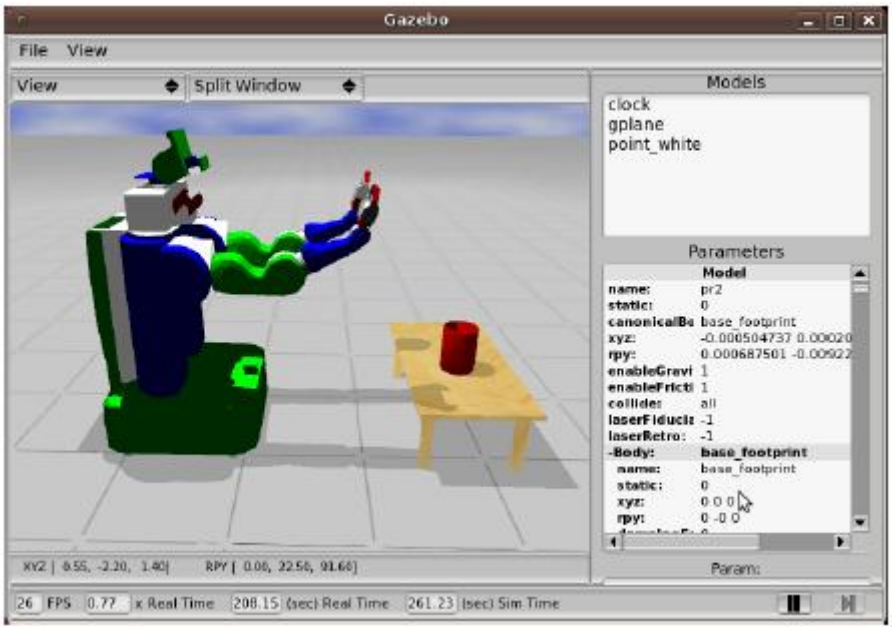

Figura 26: Simulador Gazebo

4.2.3 OpenHRP3

OpenHRP3 (centrado en el humano Arquitectura Abierta Plataforma Robótica versión 3) es una plataforma integrada de software para simulaciones de robots y desarrollos de software. Permite a los usuarios inspeccionar un modelo de robot original y programa de control de la dinámica de la simulación. Además, OpenHRP3 proporciona varios componentes de software y bibliotecas de cálculo que puede ser utilizado para robótica relacionados desarrollos de software.

OpenHRP3 (ver figura 27) se desarrolla como uno de los proyecto complementario denominado "simulador de robot tipo de componente distribuido", ejecutado por la "Cooperación de Próxima Generación Robots" que pertenece a la "Cooperación Gobierno de Ciencia y Tecnología". Motor de cálculo de dinámica está casi desarrollado por "Nakamura Lab, Departamento de mecano Informática de la Universidad de Tokio" y la interfaz gráfica se realiza por "General Robotix, Inc". Las otras partes se desarrollan como una obra de cooperar "Humanoid Resarch Group" y "Grupo de Inteligencia del Grupo de Investigación" en "Intelligent Systems Research Institute", "Instituto Nacional de Avanzada Indrustrial Ciencia y Tecnología (AIST)".

Y aunque el proyecto "simulador de robot tipo de componente distribuido" ha terminado, AIST está continuando el proceso de desarrollo como parte de en vías de subdesarrollo Plataforma OpenRT que vinculó con "proyecto de desarrollo de la tecnología inteligente para los robots de próxima generación".

Esta versión 3 (la llamada OpenHRP3) se mejora considerablemente el desarrollo en comparación con el de la distribución previa de la versión OpenHRP 2. Además, OpenHRP3 distribuye como software de código abierto. [11]

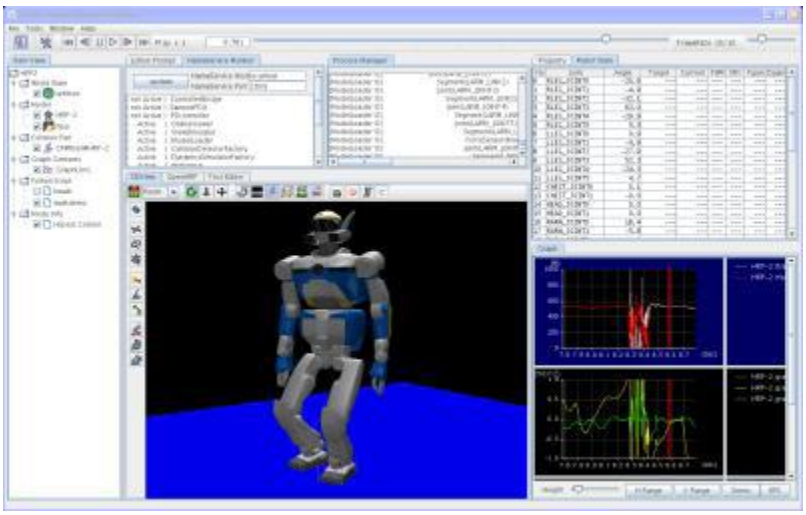

Figura 27: Simulador OpenHrp3

4.2.4 Marilou Robotics Studio

Marilou Robotics Studio (ver figura 28) fue desarrollado por la empresa ANYKODE, fu desarrollado para ayudar a personas a acelerar sus proyectos de desarrollo de la robótica.

Marilou ofrece en robótica que desean trabajar con las herramientas de un entorno de modelado y simulador de escena completa, completa con los robots con sujeción a las leyes de la física de simulación. Marilou ofrece una amplia gama de posibilidades diseñadas para que pueda crear rápidamente modelos para las entidades físicas dentro de escenas y simular sus reacciones en varias configuraciones.

Es un modelador y ambiente de simulación 3D para robots móviles, humanoides, brazos articulados y robots paralelos que funcionan en condiciones verdaderas, respetando las leyes físicas.

En un ambiente realista gráfico, Marilou nos permite crear la jerarquía necesaria para construir y probar ensamblajes de formas simples (cajas, esferas, cilindros, superficies, redes de elevación) y formas complejas, tales como triángulo de malla, con geometrías y formas convexas, posee una simulación en tiempo real o simulación acelerada.

La programación del robot se construye bajo los idiomas C, C++ y C#. en Windows y Linux. [12]

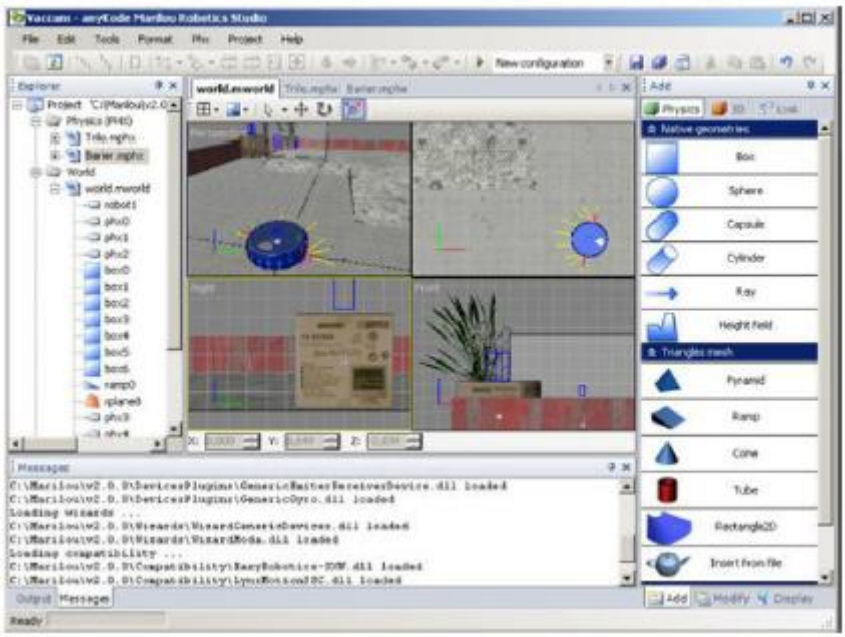

Figura 28: Simulador Marilou Robotic Studio

4.2.5 Microsoft Robotics Developer Studio 4

Microsoft Robotics Developer Studio, fue desarrollado por Microsoft, es una herramienta de programación visual para crear y depurar aplicaciones robóticas. El desarrollador puede interactuar con los robots mediante interfaces basadas en web o en Windows.

La simulación realística está provista por el motor PhysX de AGEIA. Se posibilita la emulación por software o la aceleración por hardware.

El desarrollador puede acceder fácilmente a los sensores y actuadores de los robots, proporcionada por una librería de implementación de concurrencia basada en .NET. La comunicación estás basada en mensajes, permitiendo la comunicación entre módulos.

La propia aplicación es multi-plataforma-robótica. Se permiten varios lenguajes - lenguajes Microsoft Visual Studio Express languages (Visual C#® y Visual Basic® .NET), JScript® y IronPython 1.0 Beta 1, y lenguajes de terceras partes que se adecuen a la arquitectura basada en servicios. [13]

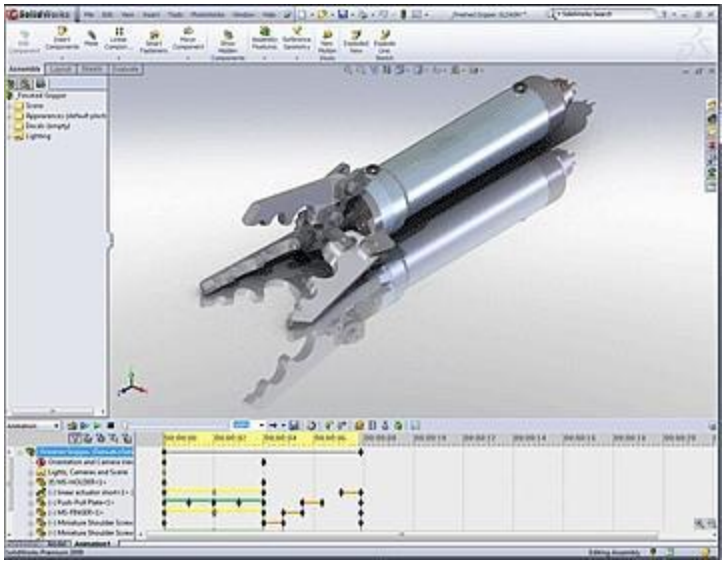

Figura 29: Simulador Microsoft Robotics Developer Sudio 4

4.2.6 OpenRAVE

OpenRAVE fue fundada por Rosen Diankov en la Calidad de Vida en el Centro de Tecnología del Instituto de Robótica de la Universidad Carnegie Mellon. Fue inspirado desde el simulador RAVE James Kuffner había comenzado a desarrollar en 1995 y utilizado por una gran cantidad de sus experimentos. El proyecto OpenRAVE se inició en 2006 y comenzó como una reescritura completa del RAVE para soportar plugins. Rápidamente se separó en su propio concepto de la arquitectura y comenzó a ser apoyado por muchos investigadores de robótica de todo el mundo. Después de obtener su doctorado en el Instituto de Robótica en agosto de 2010, Rosen Diankov convirtió en un post-doctorado en el Laboratorio de Robótica JSK en la Universidad de Tokio, donde se mantiene actualmente OpenRAVE.

Abrir Entorno Virtual Robotics Automation (OpenRAVE) proporciona un entorno para pruebas, desarrollo, y los algoritmos de planificación de movimiento en el despliegue de aplicaciones de robótica en el mundo real. La atención se centra en la simulación y el análisis de la información cinemática y geométrica relacionada con la planificación de movimientos. Naturaleza independiente de OpenRAVE le permite ser integrado fácilmente en los sistemas existentes de robótica. Proporciona muchas herramientas de línea de comandos para trabajar con robots y planificadores, y el núcleo en tiempo de ejecución es lo suficientemente pequeño para ser utilizado dentro de los controladores y los marcos más grandes.

OpenRAVE (ver figura 30) está dirigido a aplicaciones del mundo real robot autónomo, e incluye una integración perfecta de 3-D de simulación, visualización, planificación, secuencias de comandos y control. Una arquitectura de plug-in permite a los usuarios escribir controladores personalizados o ampliar la funcionalidad. [14]

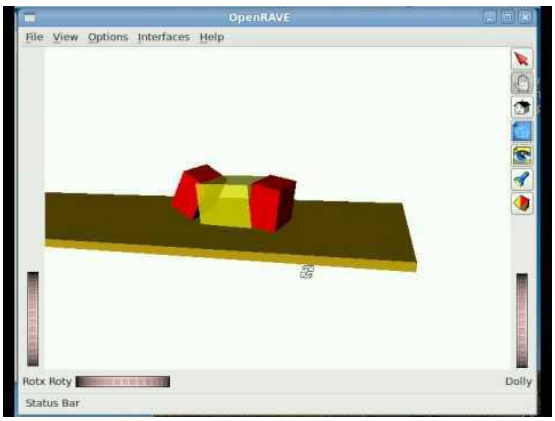

Figura 30: Simulador OpenRave

4.2.7 Webots

Webots es un software para simular robots móviles, usado con fines educativos. El proyecto Webots fue iniciado en 1996 por el Dr. Oliver Michel en el instituto federal Suizo de Tecnología EPFL en Lausanne. Una de sus principales ventajas es que permite al usuario interactuar con el modelo durante la simulación.

Webots hace uso de ODE (Open Dynamics Engine) para la detección de colisiones y simulación dinámica del cuerpo rígido. La biblioteca ODE permite simular la física de los objetos.

El programa webots permite construir robots a través de la definición geométrica y dinámica de las partes que lo componen. Igualmente permite especificar colores y texturas para una mejor visualización.

Igualmente incluye una cantidad de sensores y actuadores de uso frecuente en robótica, con sus respectivos modelos dinámicos.

El control del robot puede ser escrito en C, C++, Java, [Phyhon] y Matlab. Los modelos de robots AIBO, NAO y E-puck pueden también ser programados en URBI con la respectiva licencia.

Webots ofrece la posibilidad de tomar 'screen shots' en formato PNG y grabar simulaciones en formato MPEG o AVI. Webots guarda los modelos en un archivos .wbt, los cuales están basado en el lenguaje VRML. Es posible exportar los modelos de .wbtr a VRML al igual que objetos. [15]

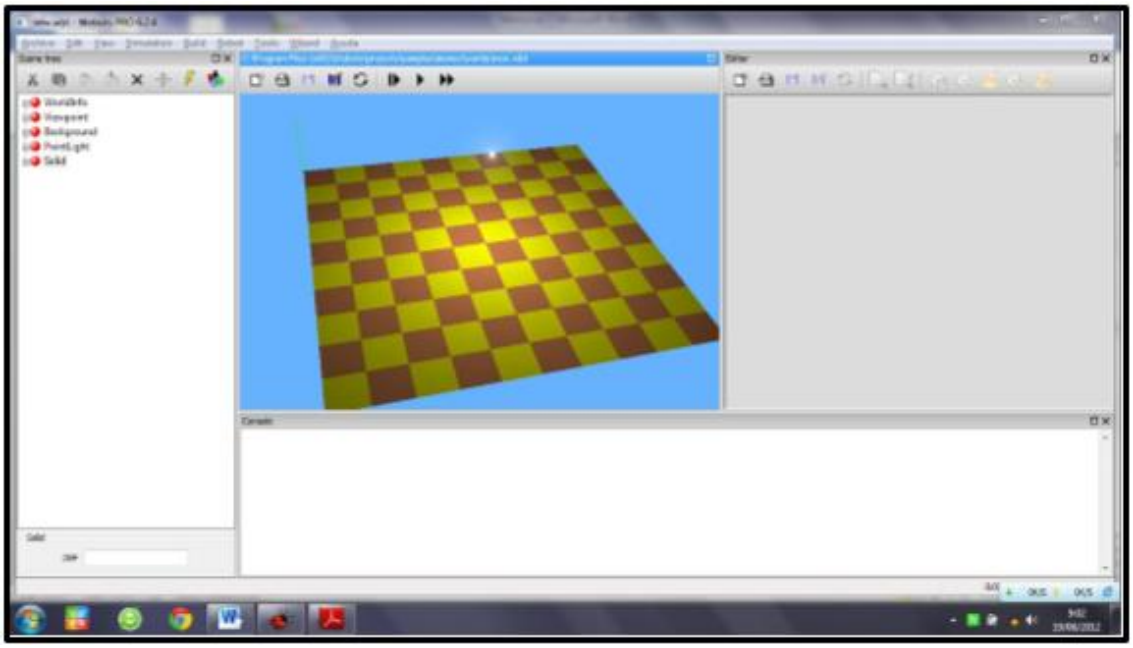

Figura 31: Simulador Webots

5 Simulador V-REP

5.1 Descripción general

El simulador de robot V-REP, con el entorno de desarrollo integrado, se basa en una arquitectura de control distribuido: cada objeto / modelo se pueden controlar individualmente a través de una secuencia de comandos incrustada, un plugin, un nodo de ROS, un cliente de API remota o una solución personalizada. Esto hace que V-REP muy versátil e ideal para aplicaciones multi-robots. Los controladores pueden ser escritos en C / C + +, Python, Java, Lua, Matlab o Urbi. V-REP se utiliza para el desarrollo de algoritmos rápidos, simulaciones de automatización de fábricas, prototipado rápido y la verificación, la educación relacionada con la robótica, control remoto, seguridad de doble control, etc. V-REP es gratuito para los estudiantes, dado que utilizan el software en casa. [16]

Un simulador de robótica se utiliza para crear aplicaciones embebidas para un robot sin depender físicamente en la máquina real, con el consiguiente ahorro de costes y tiempo. En algunos casos, estas aplicaciones pueden ser transferidos en el robot real (o reconstruidas) sin modificaciones. El término simulador robótica puede hacer referencia a varias aplicaciones de simulación robótica diferentes. Por ejemplo, en robótica móvil aplicaciones, robótica basados en el comportamiento simuladores permiten a los usuarios crear mundos simples de objetos rígidos y fuentes de luz y para programar robots para interactuar con estos mundos. Simulación basada en el comportamiento permite acciones que son más de naturaleza biológica en comparación con los simuladores que son más binarios o computacionales. Además, los simuladores basados en el comportamiento pueden "aprender" de los errores y son capaces de demostrar la antropomorfocalidad de tenacidad.

Una de las aplicaciones más populares para los simuladores de la robótica es para el modelado 3D y diseño de un robot y su entorno. Este tipo de software de robótica tiene un simulador que es un robot virtual, que es capaz de emular el movimiento de un robot real en un sobre de trabajo real. Algunos simuladores robóticos, utilizan un motor de física para la generación de movimiento más realista del robot. El uso de un simulador de robótica para el desarrollo de un programa de control de robótica es muy recomendable independientemente de si un robot real está disponible o no. El

simulador permite a los programas de robótica ser convenientemente escritas y depuradas fuera de línea con la versión final del programa probado en un robot real. Por supuesto, esto es principalmente para robóticos industriales sólo las aplicaciones, ya que el éxito de la programación fuera de línea depende de cómo similar es el entorno real del robot para el entorno simulado sensores basados en acciones del robot son mucho más difíciles de simular y para programar fuera de línea, ya que el movimiento del robot depende de las lecturas del sensor instantáneos en el mundo real. [17]

En la figura 32 se muestra la estructura jerárquica principal de este simulador.

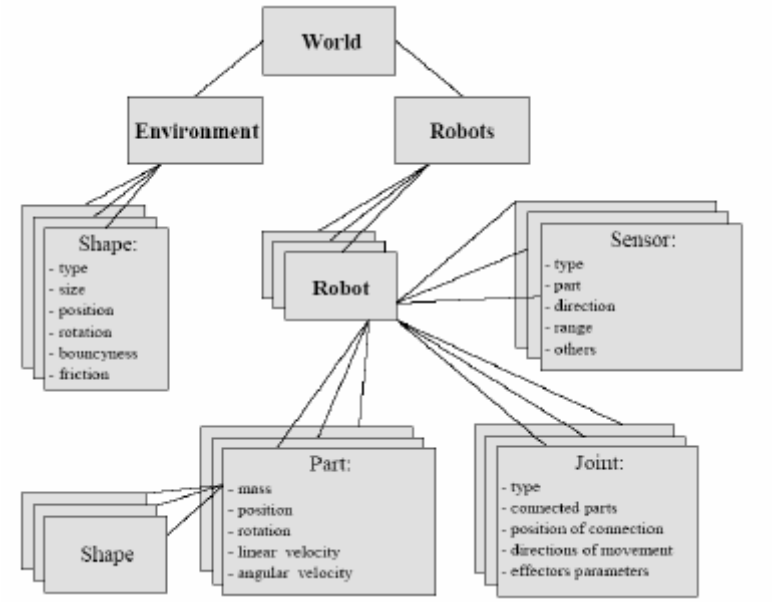

Figura 32: Estructura Jerárquica principal

5.2 Interfaz

V-REP ha sido desarrollado por Coppelia Robotics. Para la realización de este PFC hemos utilizado la versión 3.0.1 de V-REP que corre bajo el sistema operativo Windows 7 (ver figura 33). La interfaz de usuario está compuesta por las siguientes ventanas, la ventana principal de simulación que está en la parte más derecha en la misma aparecen todos los elementos que componen una escena de simulación, un poco más a la izquierda aparece la ventana de la escena de jerarquía en donde van a aparecer

desplegados cada uno de los elementos que estén presentes en la ventana de simulación así como la relación jerárquica que pueda aparecer entre los mismos. Y la ventana que está más a la izquierda es el explorador de modelos, un modelo es un objeto propio de v-Rep. en el cual se encapsulan las funcionalidades de un elemento construido en v-Rep. y que después se pueden visualizar utilizando esta ventana entre los modelos ya cargados aparecen robots de base fija y de base flotante entre otros.

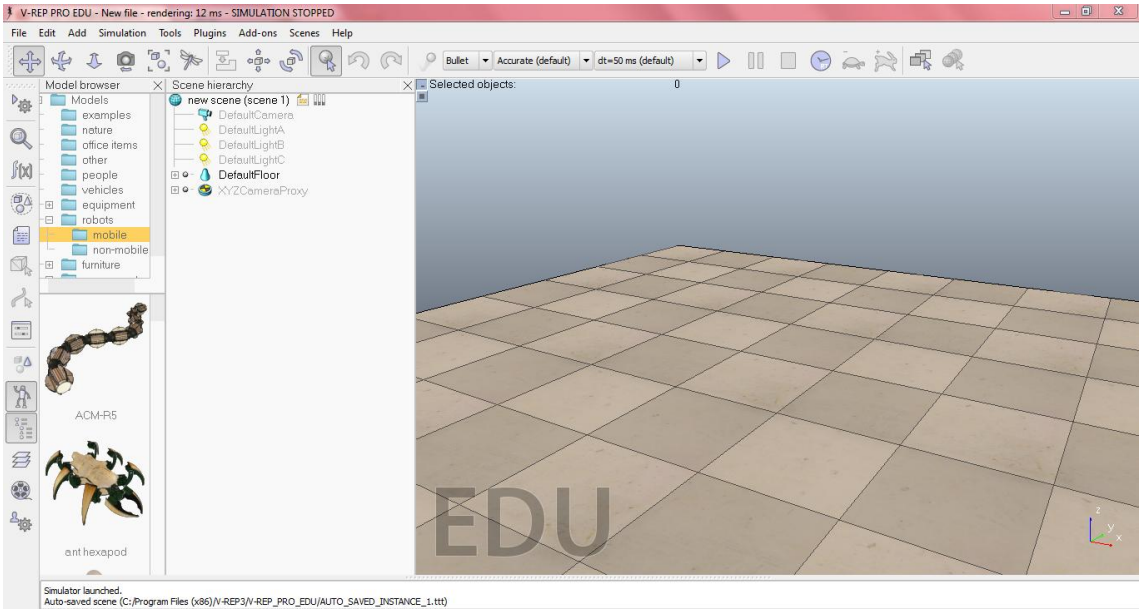

Figura 33: Interfaz V-Rep

V-Rep cuenta con dos barras de herramientas, una horizontal y otra vertical (ver figura 34). En la horizontal aparecen ítems de navegación y exploración de elementos. Y la vertical está conformada por botones que son enlaces a las funcionalidades principales de v-Rep.

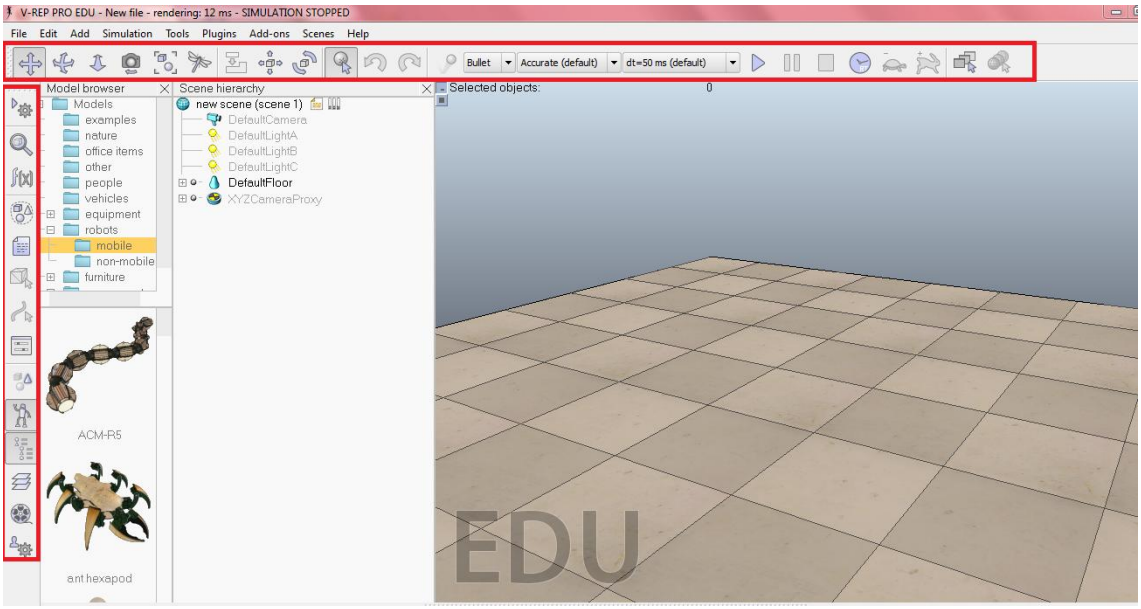

Figura 34: Herramientas principales V-Rep

Para visualizar mejor estas funcionalidades arrastramos el manipulador que vamos a emplear a la ventana principal (ver figura 35). Modelo que ha sido previamente diseñado mediante la herramienta Solidworks, tomando como referencias el robot ya existente en la universidad Carlos III de Leganés.

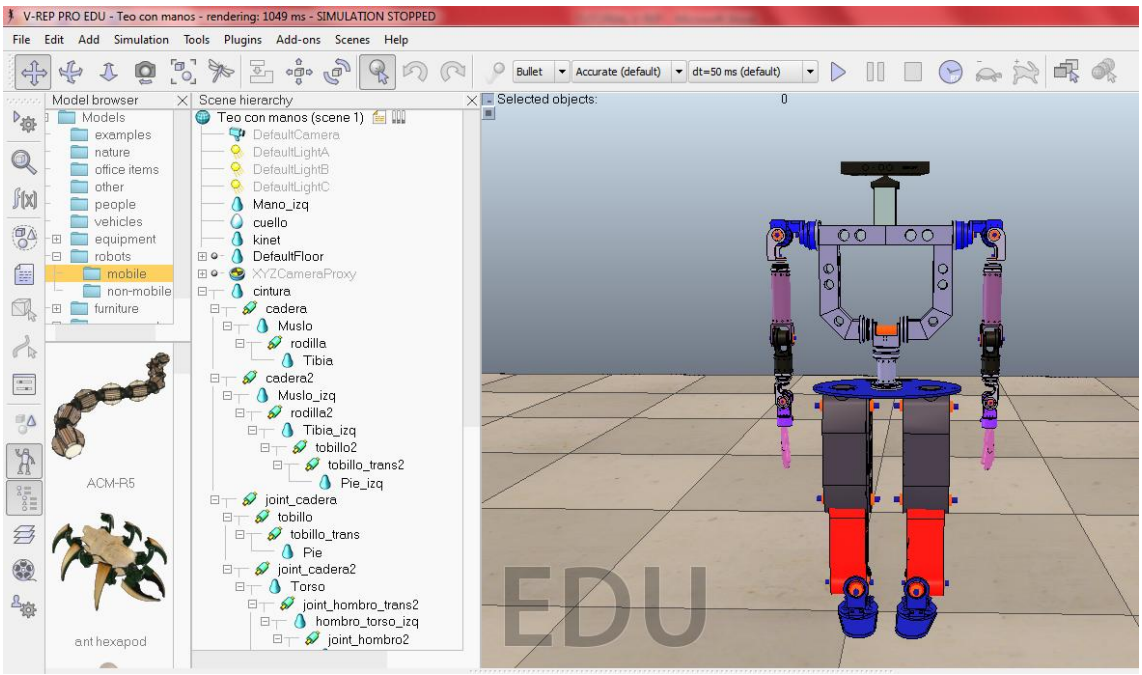

Figura 35: Pantalla principal

La barra de herramientas superior también cuenta con unos botones de vistas llamados “pages” a través de los cuales podemos ver la escena de simulación desde varias vistas. En la ventana de jerarquía podemos ver que está conformado el robot y la relación que existe entre sus elementos (ver figura 36).

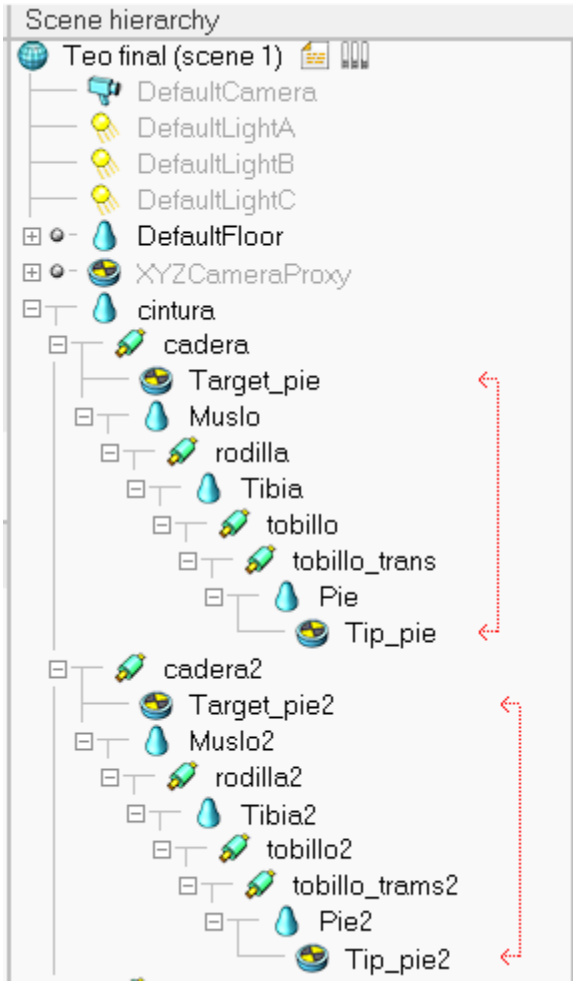

Figura 36: Ventana jerárquica del robot

5.3 Creación de objetos

Para la creación de objetos vamos a la opción “Add” del menú principal en donde podemos observar todos los elementos que podemos agregar a una escena de simulación entre ellos se encuentra el ítem “Primitive Shape” que no despliega todas las primitivas básicas que podemos crear como por ejemplo un cilindro (ver figura 37).

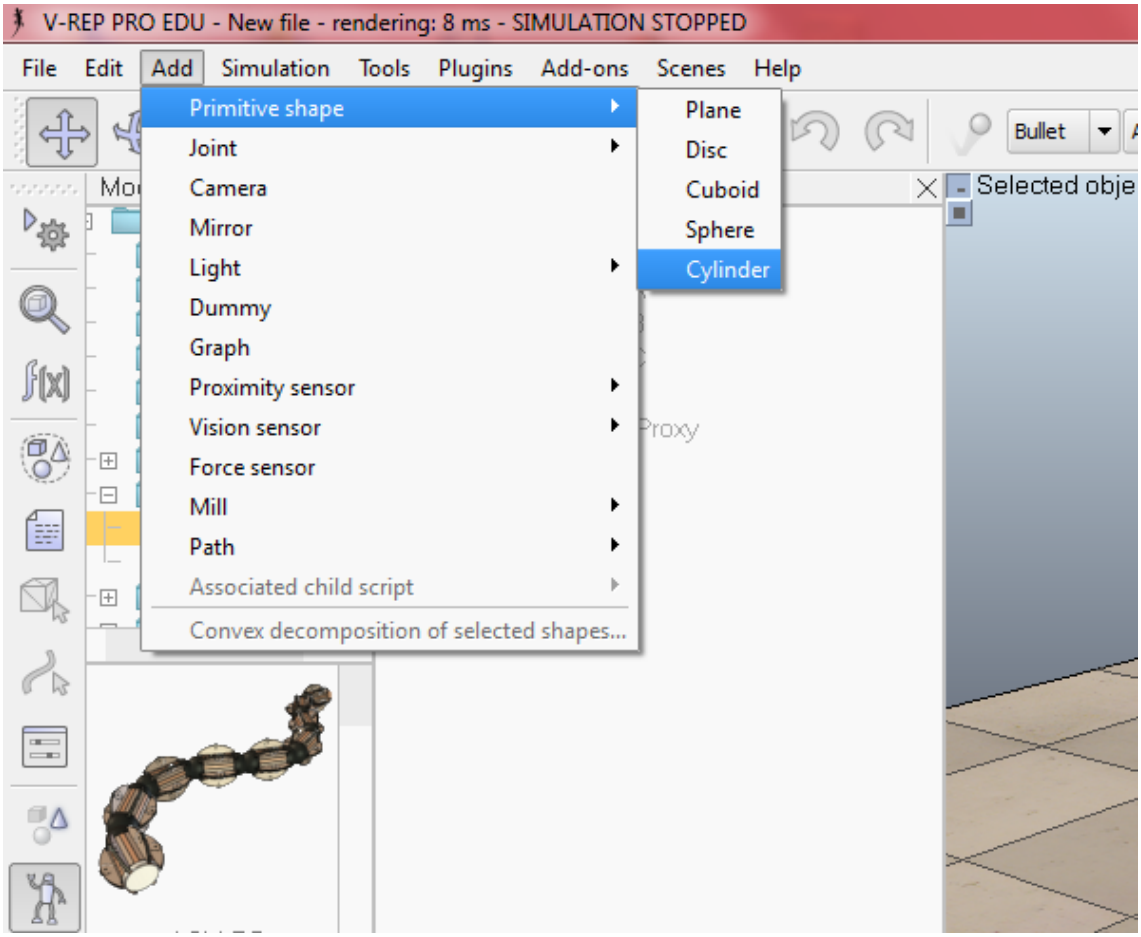

Figura 37: Crear objetos

Una vez lo seleccionamos nos aparece una caja de dialogo en donde podemos modificar las propiedades geométricas iniciales de la forma que vamos a adicionar una vez tengamos la forma en la escena de simulación, podemos editarla haciendo doble clic en el icono que no aparece en la parte izquierda del ítem que representa la figura en la jerarquía de la escena. Se despliega el menú “Shape Properties” que nos permite editar tanto las propiedades geométricas como dinámicas de la figura 38 en cuestión, propiedades tales como los componentes del color la geometría y la masa y el tensor de inercia del elemento en cuestión.

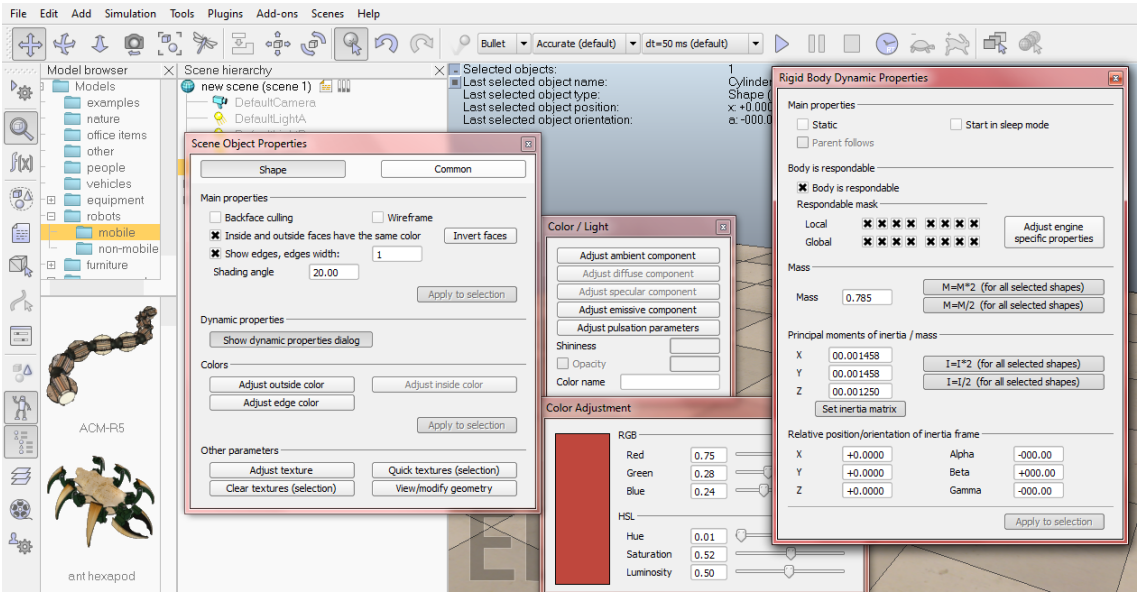

Figura 38: Propiedades de los objetos

Adicionalmente mediante la opción “Edit Modes” del submenú “Edit” podemos elegir entre tres modos de edición que nos permite obtener figuras aún más complejas con estas opciones podemos editar los vértices los lados y los triángulos que conforman un elemento, para este detalle de los elementos, hemos preferido usar la herramienta Solidworks para definir y construir elemento a elemento del robot, y después exportarlo a V- Rep. donde se hace el ensamblaje, ya que Solidworks nos permite editar y construir con más detalles los elementos del robot.

5.4 Importar objetos

Por lo general las piezas que conforman el eslabón de un robot son un objeto tipo STL previamente desarrollado mediante la herramienta de diseño SolidWorks.

STL (siglas provenientes del inglés "'STereo Lithography'"') es un formato de archivo informático de diseño asistido por computadora (CAD) que define geometría de objetos 3D, excluyendo información como color, texturas o propiedades físicas que sí incluyen otros formatos CAD.

Fue creado por la empresa 3D Systems, concebido para su uso en la industria del prototipado rápido y sistemas de fabricación asistida por ordenador. En especial desde los años 2011-2012 con la aparición en el mercado de impresoras 3D de extrusión de plástico termofusible (personales y asequibles), el formato STL está siendo utilizado ampliamente por el software de control de estas máquinas.

Un archivo STL describe una desestructurada prima triangular la superficie por la unidad de lo normal y vértices (ordenado por la regla de la mano derecha) de los triángulos utilizando una imagen tridimensional del sistema de coordenadas cartesianas . Coordenadas STL deben ser números positivos, no hay información de escala, y las unidades son arbitrarias.

En V-REP, los objetos se encuentran compuestos por uno o varios nodos. Estos nodos son descritos utilizando formato .STL importados así de solidworks para poder manejarlos en V-REP y ensamblar todos los nodos y formar en el simulador V-REP el robot TEO, un nodo es la estructura mínima indivisible y tiene como misión la de definir las características de un objeto o bien las relaciones entre distintos objetos.

Los nodos a su vez contienen campos que describen propiedades. Todo campo posee un tipo determinado y no se puede inicializar con valores de otro tipo. De este modo, cada tipo de nodo tiene una serie de valores predeterminados para todos sus campos, de forma que cuando se utilice en una escena sólo se deben indicar aquellos campos que se quieran modificar.

Para importar los elementos seleccionamos lo opción “Import Binary STL” del submenú “File” y cargamos el objeto (ver figura 39), hay que importar objeto por objeto y una vez importados en la escena principal con las funciones que ya hemos explicado de cinemática y geometría podemos ensamblar los eslabones del robot.

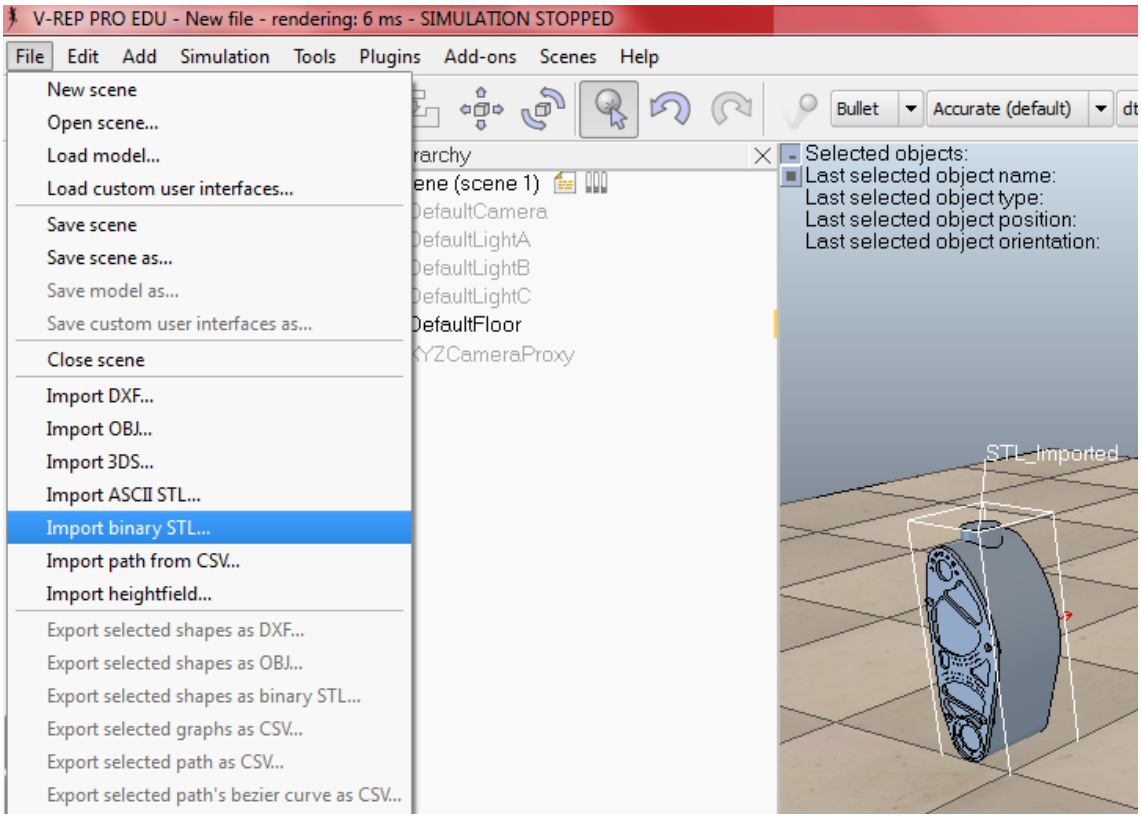

Figura 39: Objetos importados de Solidworks

5.5 Creación de eslabones

Para la creación de los eslabones del robot (añadir nombre del robot), estos han sido previamente construidos mediante la herramienta Solidworks al diseñar los objetos de cada elemento del robot y guardados con el formato .STL, por esto hacemos clic en la opción “File” “Import Binary STL” y cargamos el primer eslabón del robot y así sucesivamente (ver figura 40), una vez cargados todos los eslabones podemos tener una vista un poco más detallada de los mismos utilizando la opción de acercamiento que se activa con la rueda central del ratón.

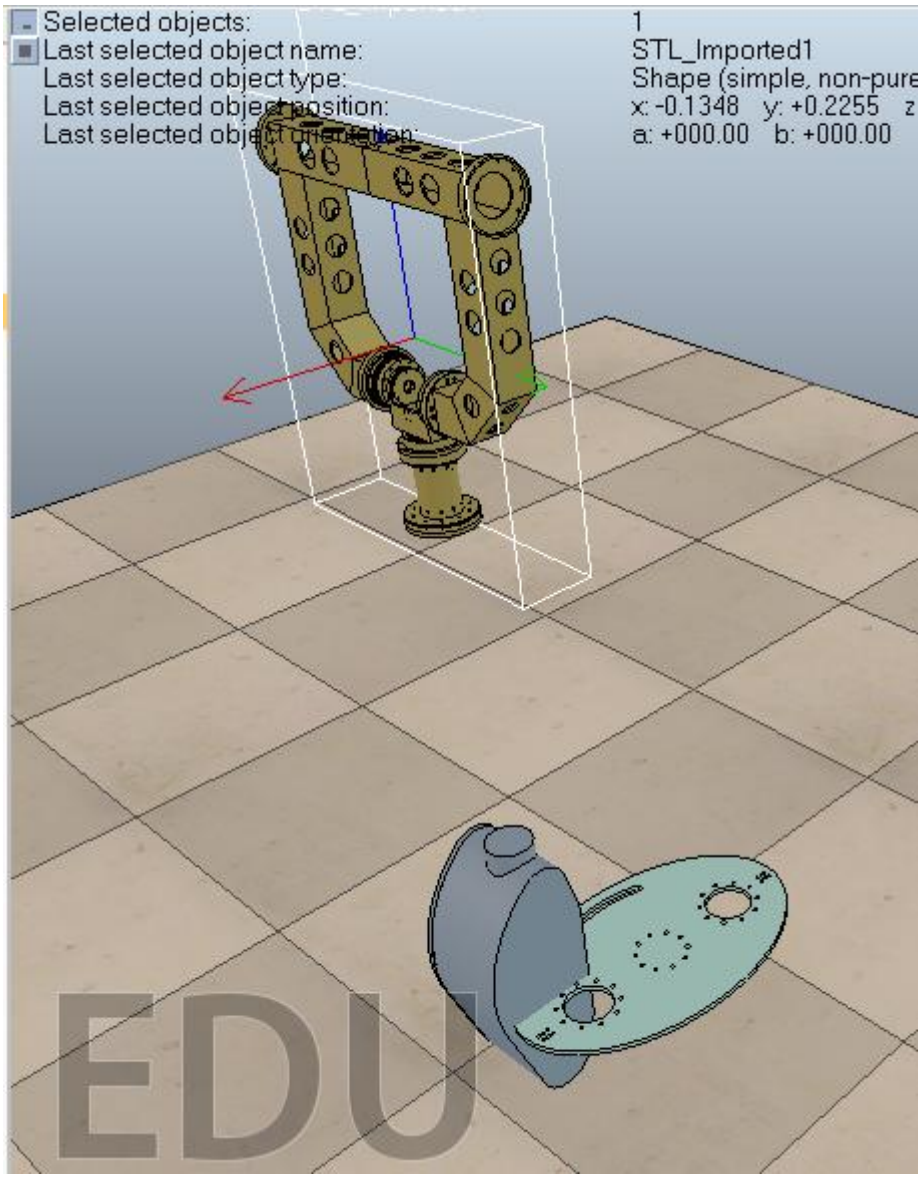

Figura 40: Creación de eslabones

En la ventana de jerarquía podemos cambiar el nombre de cada uno de los eslabones (ver figura 41), posicionándonos en el ítem respectivo de cada uno de ellos y asociarles el nombre que queramos colocar, como por ejemplo cintura, muslo, tibia y pie para un lado del robot y muslo2, tibia2, pie2 para el otro lado y así con todos los eslabones.

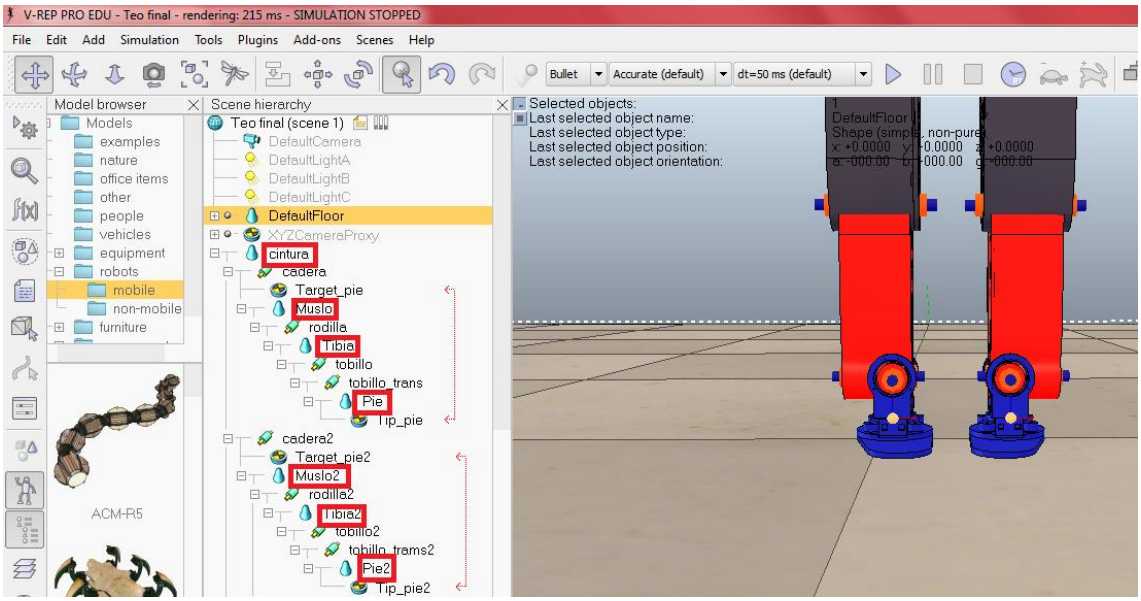

Figura 41: Asociación de los eslabones

Posteriormente con la opción "Position/Orientation" los podemos posicionar y orientar en el espacio de acuerdo a especificaciones previamente conocidas (ver figura 42), es decir, debemos conocer la morfología del robot antes de realizar esta operación. Ejecutamos esta operación tanto para todos los eslabones para ensamblar el robot.

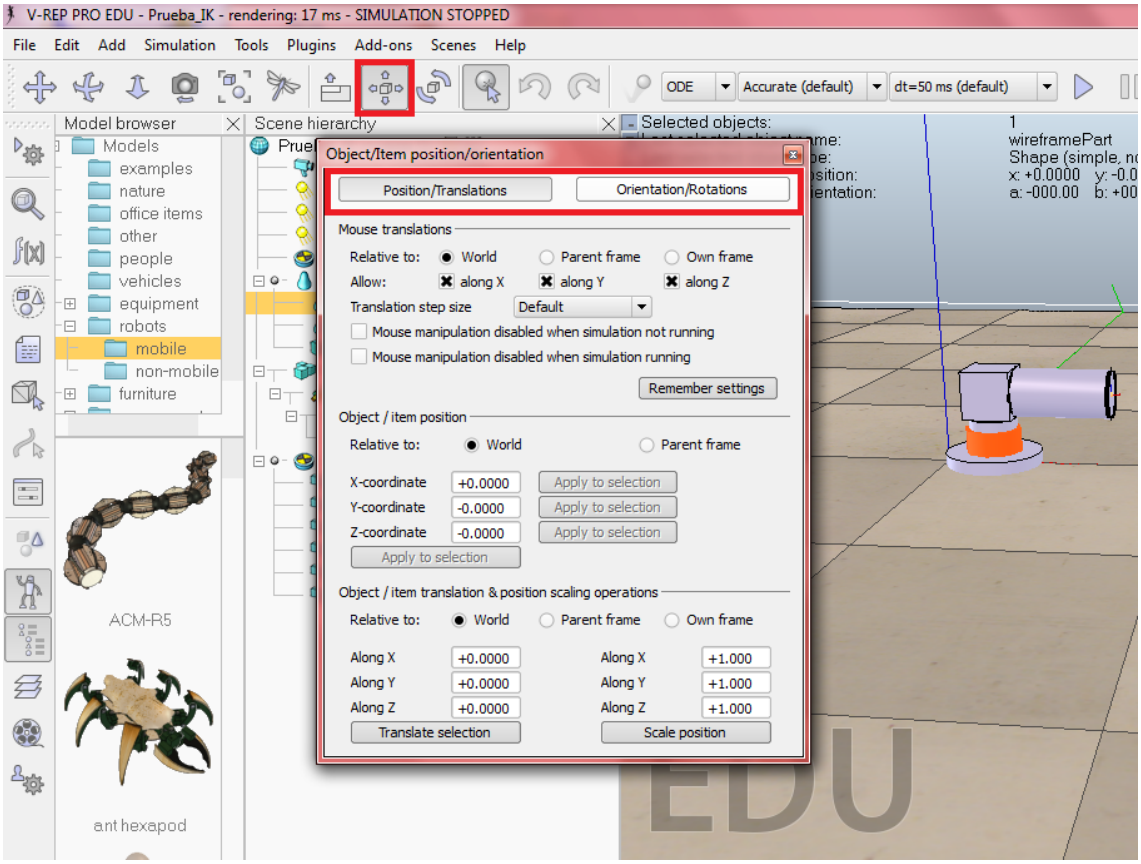

Figura 42: Posición de los eslabones

Después de aplicar el mismo procedimiento para los eslabones restantes obtenemos la representación tridimensional del robot conformada por sus elementos base y los demás eslabones de la cadena serial (ver figura 43).

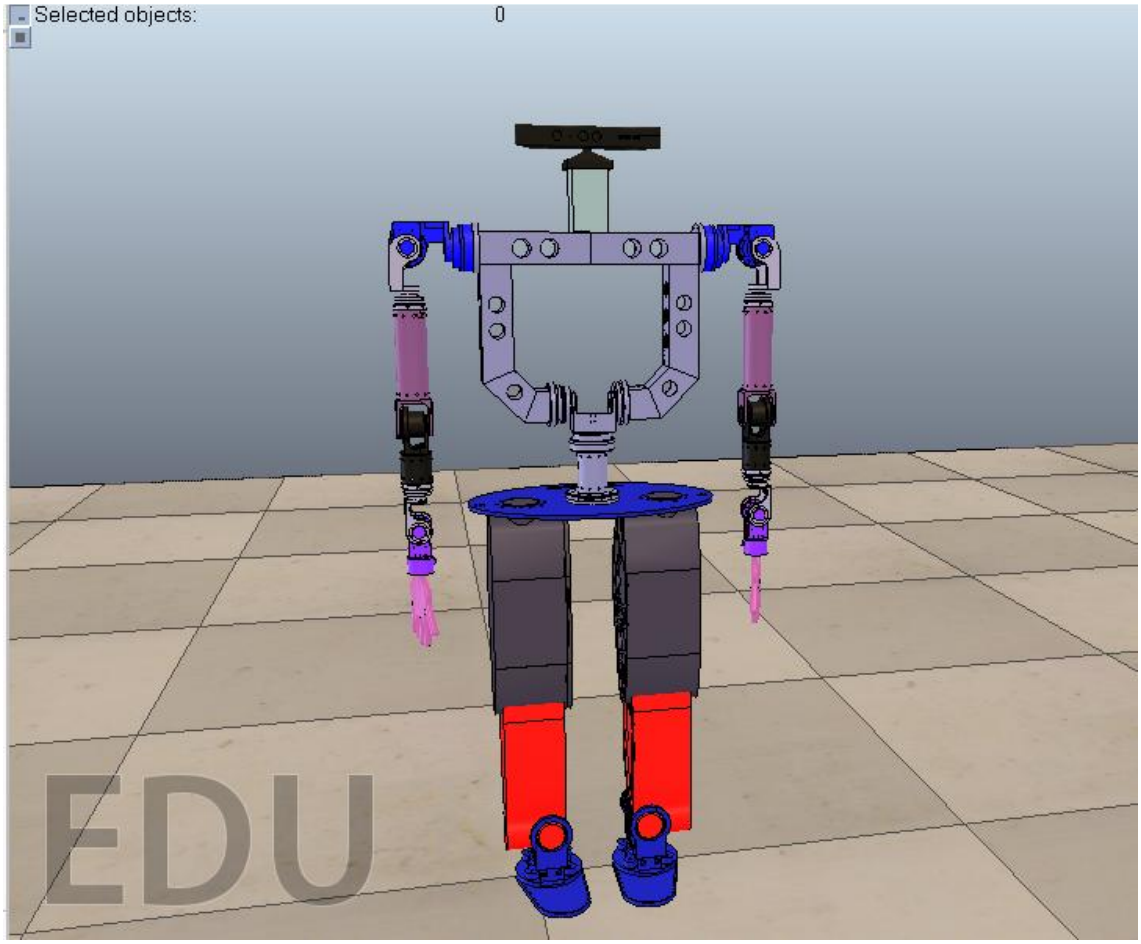

Figura 43: Representación del Robot Teo virtual

5.6 Creación de articulaciones

5.6.1 Diseño de articulaciones de un robot serial

Una vez que tengamos los eslabones de la cadena serial del robot debidamente posicionados y orientados procedemos a crear los elementos que representan las articulaciones del robot (ver figura 44) y asociarlas a su eslabón respectivo, para ello seleccionamos la opción "Add" del menú principal y presionamos sobre la opción "joint", podemos ver los tres tipos de articulaciones que soporta V-Rep, cilíndrica prismática y esférica.

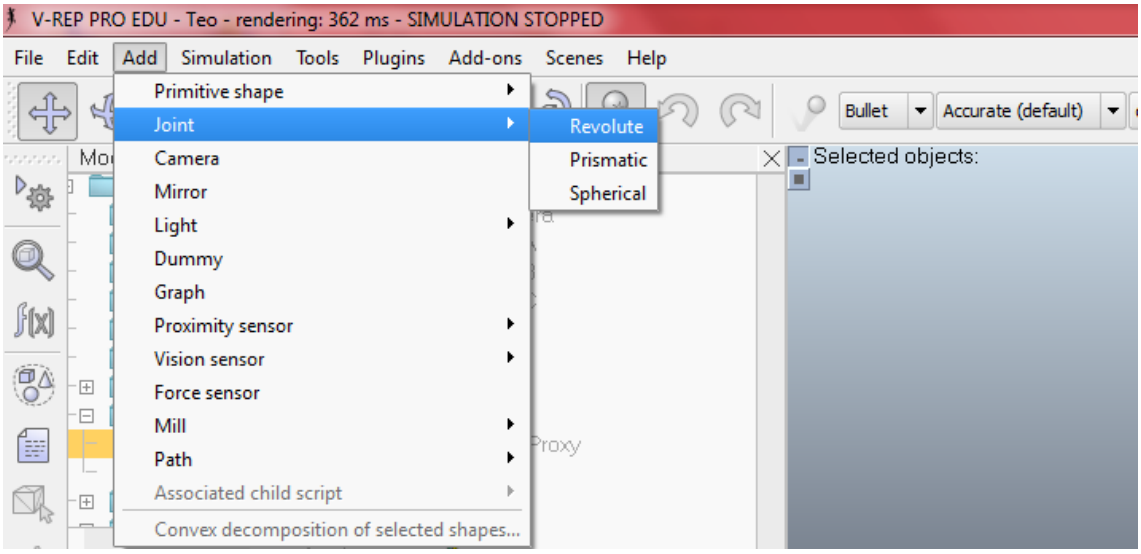

Figura 44: Diseño de articulaciones

Seleccionamos la opción cilíndrica que es la que vamos a utilizar para este robot una vez adicionamos las articulaciones, les cambiamos le nombre haciendo doble clic sobre el ítem que representa a cada una de ellas en la escena de jerarquía (ver figura 45).

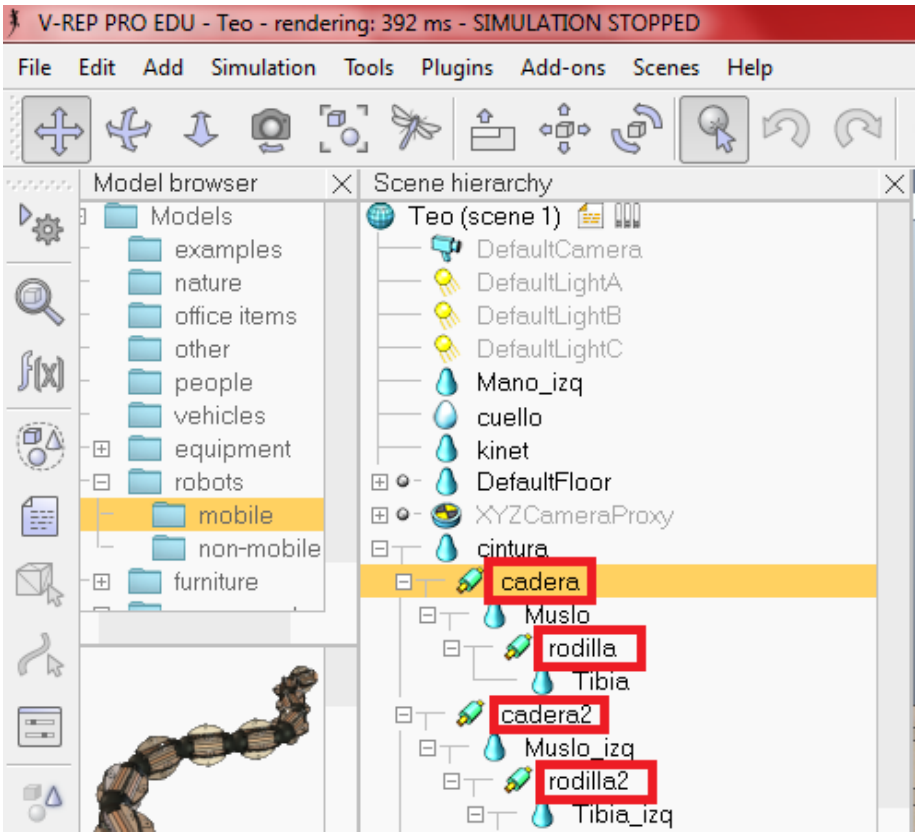

Figura 45: Articulaciones en la estructura jerárquica

Si queremos ver en donde han sido ubicadas estas articulaciones, podemos visualizar desplegando los sólidos tridimensionales en modo mallado (ver figura 46), para ello presionamos el botón derecho del ratón y sobre la ventana de simulación y el la opción "view" deshabilitamos la opción "solid rendering".

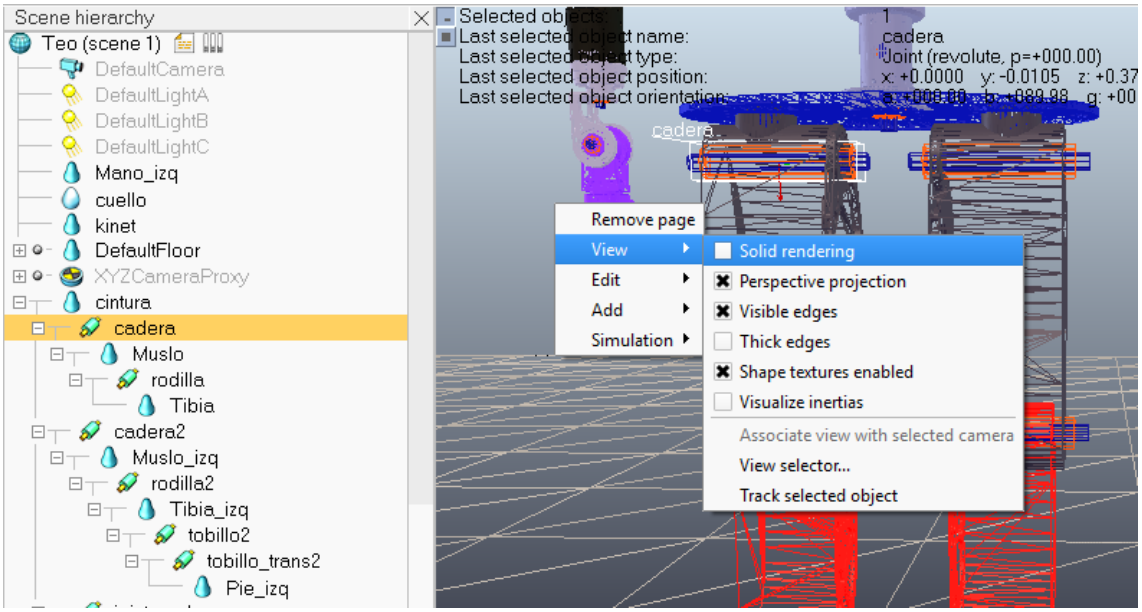

Figura 46: Articulaciones

Paso seguido tenemos que asociar cada una de las articulaciones a su eslabón respectivo, en nuestro robot seleccionamos la articulaciones y en conjunto seleccionamos el objeto al que va a asociarse cada articulación y damos clic sobre el botón “Position/Orientation” ubicado en la barra de herramientas que está en el lado izquierdo de la aplicación, nos aparece la caja de dialogo de dicha opción y seleccionamos la opción “ Apply to selection” que se encuentra justo por debajo de las coordenadas, realizamos la misma operación para las siguientes articulaciones del robot (ver figura 47) y volvemos al modo mallado para observar las nuevas posiciones de las articulaciones.

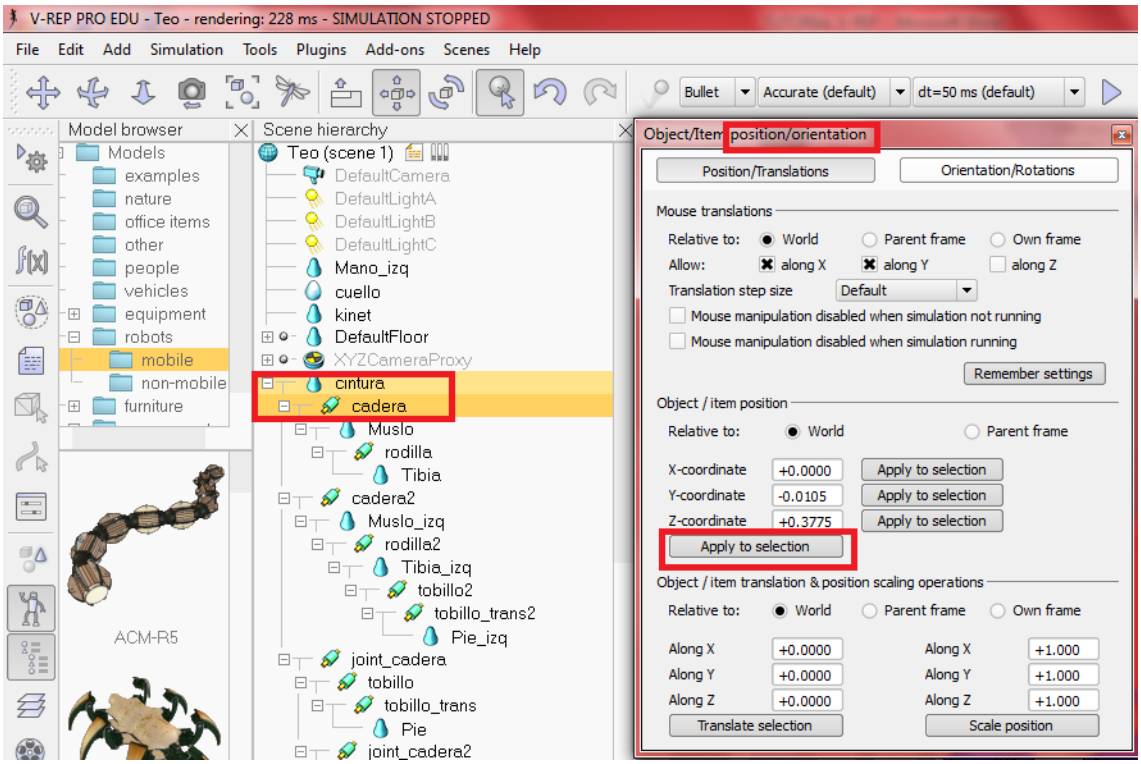

Figura 47: Articulaciones asociados a cada eslabón

5.6.2 Posicionamiento y orientación de la articulación de acuerdo a su eje de rotación

La orientación de las articulaciones se realiza con respecto al eje inercial del mundo por lo tanto las articulaciones que tiene que estar orientadas con respecto al eje “Y” del mundo hay que rotarlas 90 grados con respecto al Angulo alpha y las que tenga que estar orientadas con respecto al eje “X” las rotamos 90 grados con respecto al

ángulo beta de la articulación. Finalmente hacemos algunos cambios breves a su posición si esto es necesario (ver figura 48), todo esto lo realizamos con la caja de dialogo “Position/Orientation” para cada una de las articulaciones del robot.

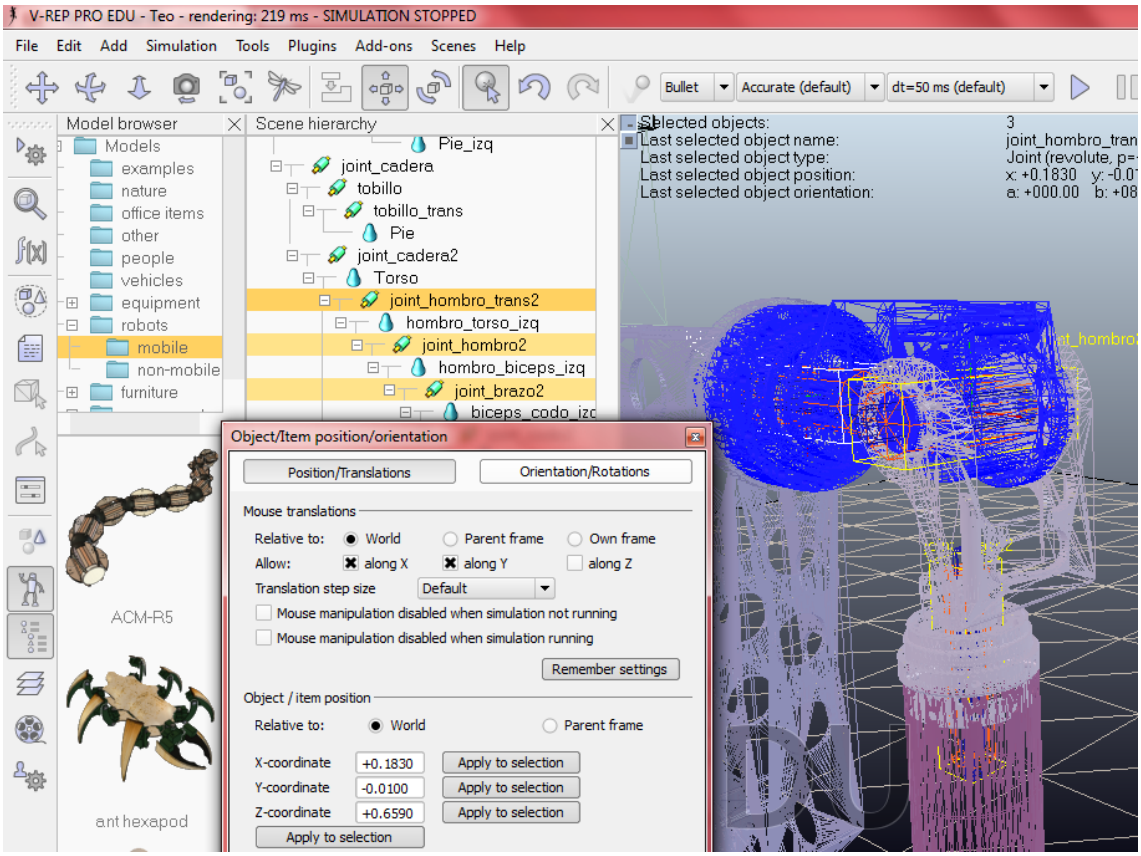

Figura 48: Posición de cada articulación

5.6.3 Creación de la cadena serial

Una vez hemos hecho esta relación para todos los eslabones y articulaciones tenemos que establecer la relación jerárquica existente entre los mismos (ver figura 49). Para ello vamos en sentido contrario, desde la última articulación del robot, seleccionando la articulación del eslabón respectiva y teniendo como último objeto seleccionado al objeto que va a ser el padre entre estos dos. Pulsamos el botón derecho del ratón y de la opción “Edit” y seleccionamos la opción “Make Last Selected Object Parent” realizamos la misma operación secuencialmente hasta llegar a la base del robot.

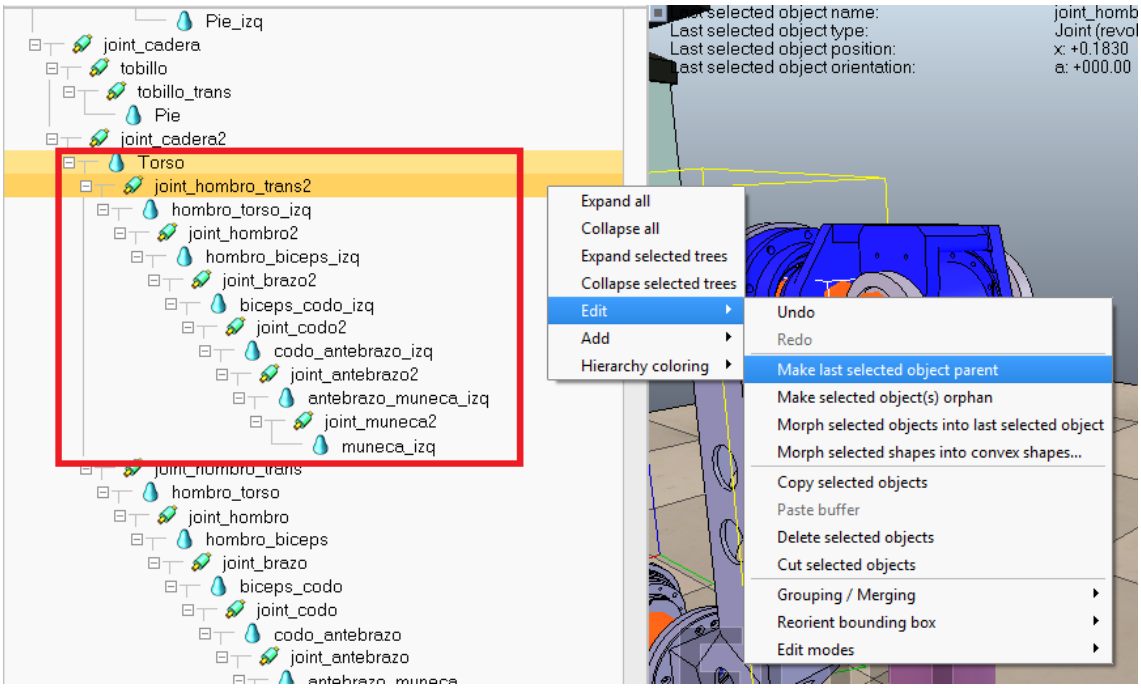

Figura 49: Cadena serial

5.6.4 Cadena serial con los elementos "Tip" y "Target"

Como último paso tenemos que adicionar los dos últimos elementos que necesita V-Rep para ver una cadena serial como una cadena cinemática, los elementos "tip" y "target", estos dos son objetos tipo "Dummy" (ver figura 50), con él defino un punto en el espacio, de la opción "Add" seleccionamos "Dummy" y lo posicionamos y orientamos en el mismo sentido en el que debe estar el efector final del robot.

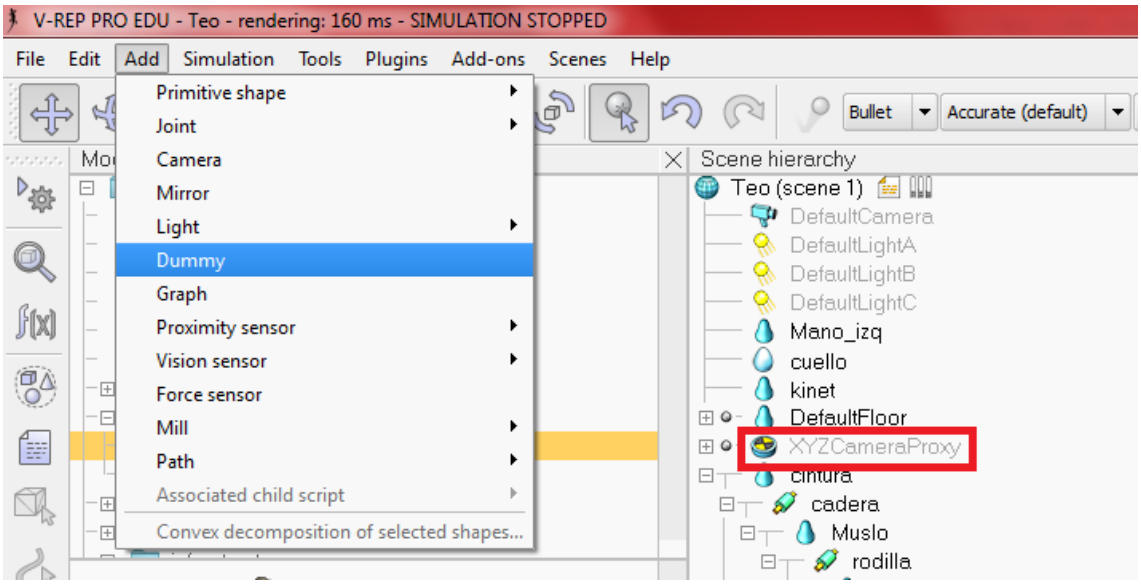

Figura 50: Creación de los elementos Tip y Target

Este objeto “tip” es el hijo de la última articulación como podemos ver en la figura 51.

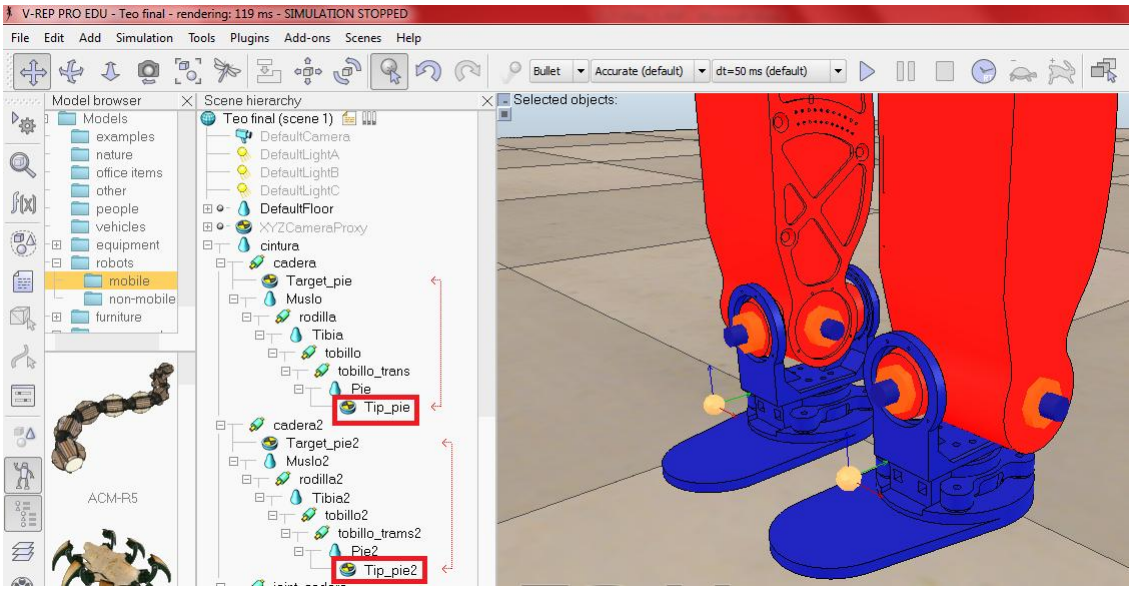

Figura 51: Objeto Tip

El objeto “target” es un punto en el espacio que va a seguir el robot cuando ejecutemos una simulación de tipo cinemática, inicialmente tiene la misma posición del objeto “tip” y relacionamos estos dos elementos, seleccionando el icono que representa el elemento “tip” en la escena de jerarquía en la opción “Linked dummy” seleccionamos “target(dummy)” y en “Link type” la opción “tip_target”, podemos apreciar que aparece una flecha punteada en doble sentido que nos indica que la relación ha sido correctamente establecida entre estos dos elementos (ver figura 52). Una vez hayamos realizado este proceso queda la cadena cinemática completamente configurada y podemos proceder a simularla cinemáticamente.

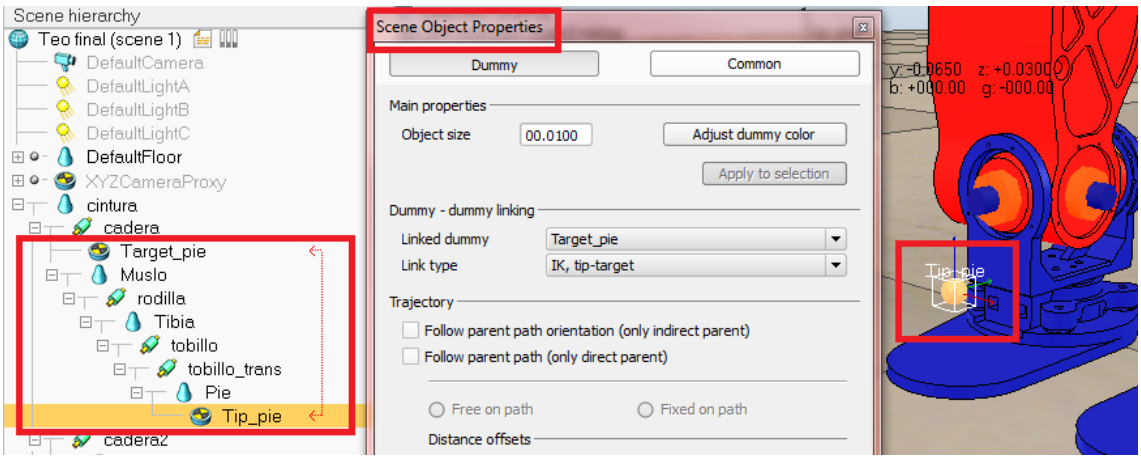

Figura 52: Objeto Target

5.7 Cinemática inversa

En la barra de herramientas ubicado en el lado izquierdo de la aplicación existe un botón llamado “Inverse Cinematics” lo seleccionamos y aparece desplegada la caja de dialogo respectiva y sus diferentes opciones, como paso inicial creamos un nuevo grupo utilizando el botón “Add new IK group” y asociamos a este grupo un elemento tip que es el que realizamos previamente (ver figura 53), adicionamos este elemento y con el botón “Add new IK element” y asociamos un elemento base que es la base del robot también creada anteriormente de cuando definimos los eslabones y articulaciones del robot.

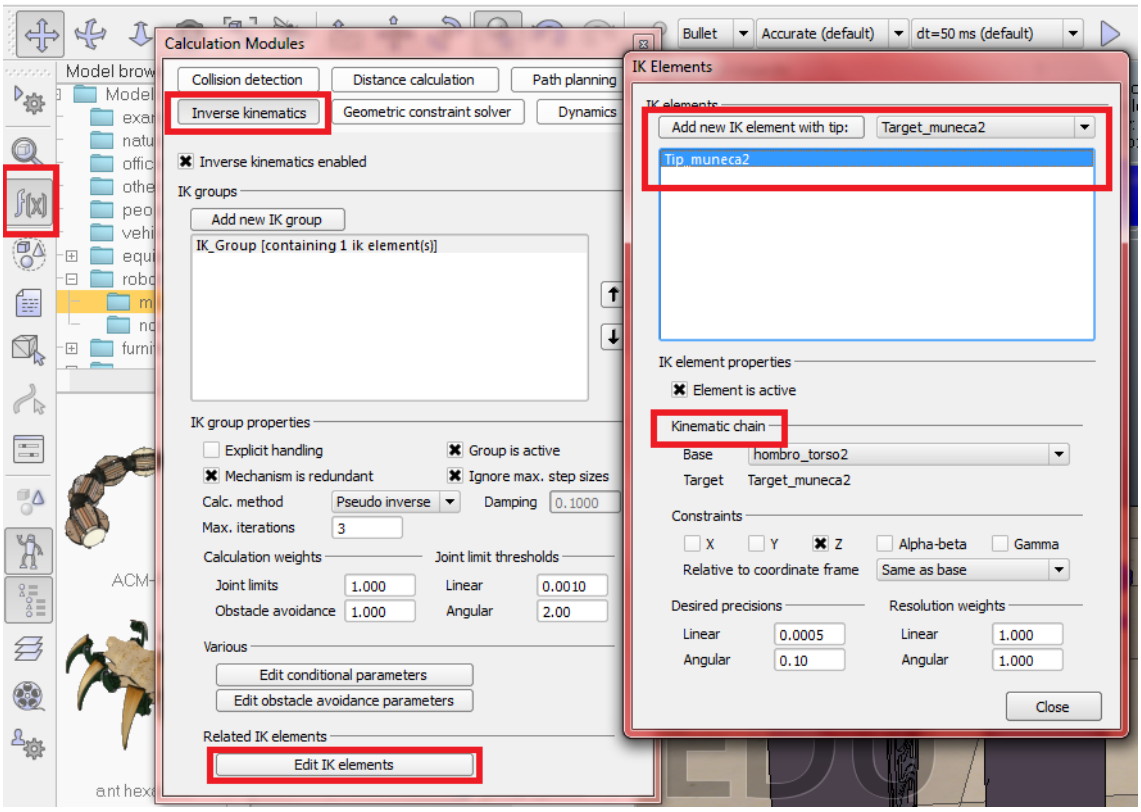

Figura 53: Creación cinemática inversa

Entre los parámetros del cálculo de la cinemática inversa se pueden apreciar los dos métodos soportados por V-Rep “Pseudo inverse” y “ DLS” para nuestro robot seleccionamos “Pseudo inverse” (ver figura 54) y lo ejes sobre los que queremos posicionar y orientar el efector final en su simulación.

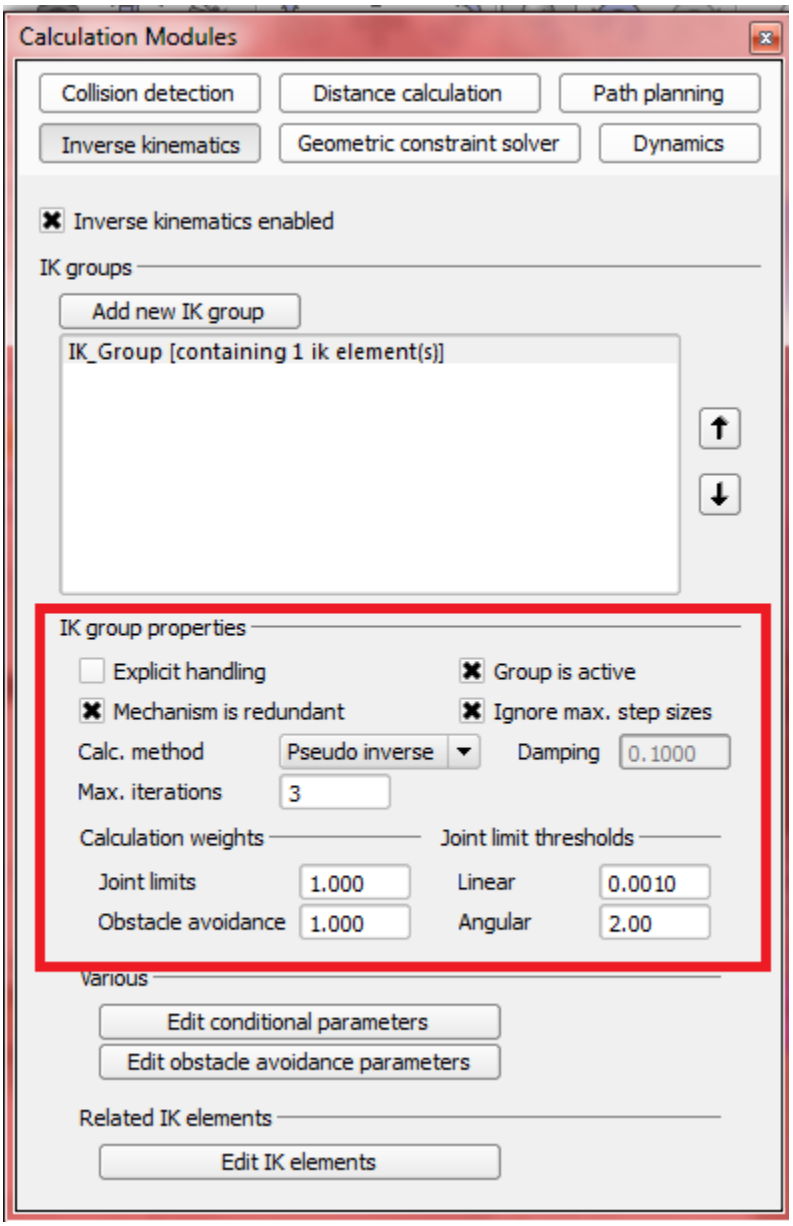

Figura 54: Parámetros de la cinemática inversa

Nos posicionamos sobre el elemento “target” ejecutamos la simulación con el botón respectivo (ver figura 55) y seleccionamos la opción “objetive en barra shifp” de la barra de herramientas y mediante esta podemos trasladar el elemento “target” utilizando el ratón y apreciar como el efector final sigue a este punto en el espacio (ver figura 56 y 57). Igualmente con el botón “object barra ítem rotative” podemos realizar los mismo pero esta vez rotando el elemento “target” y podemos apreciar como la solución de cinemática inversa genera los ángulos necesarios para que las articulaciones del robot giren de acuerdo a como giramos el elemento “target” de esta forma verificamos que la solución de cinemática inversa para nuestro robot ha quedado correctamente especificada.

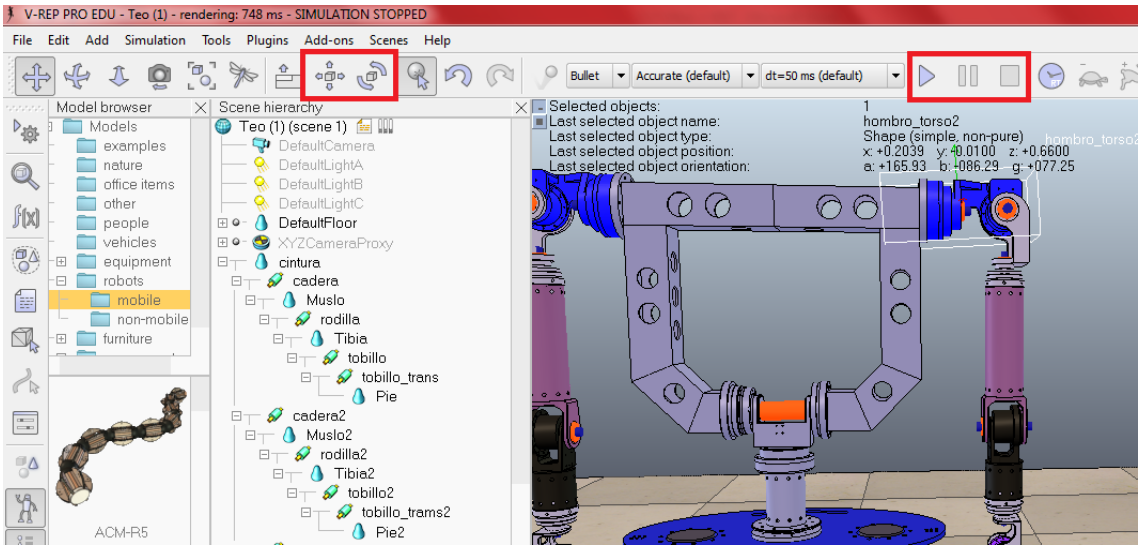

Figura 55: Simulación de la cinemática inversa

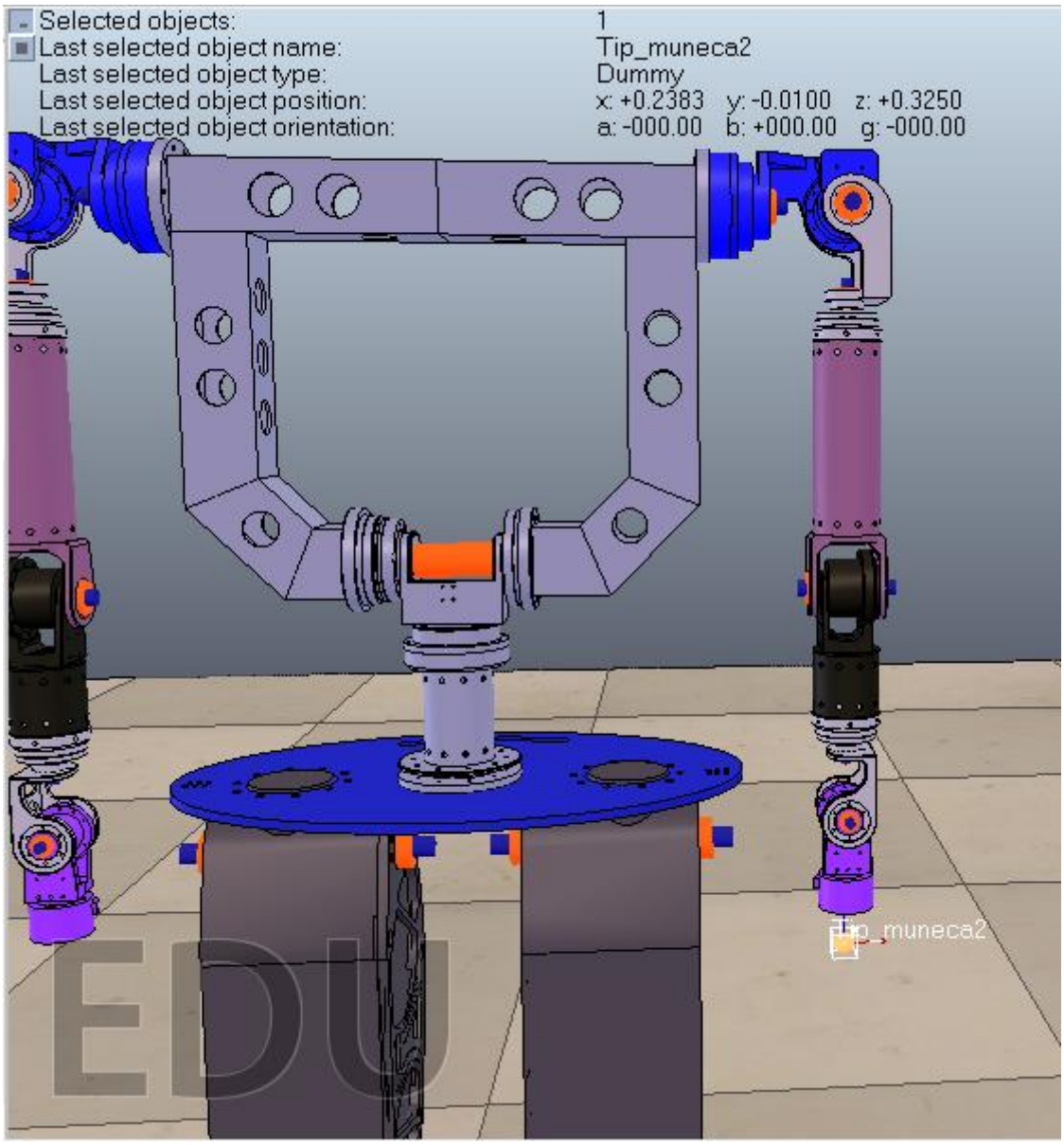

Figura 56: Simulación del brazo izquierdo de Teo

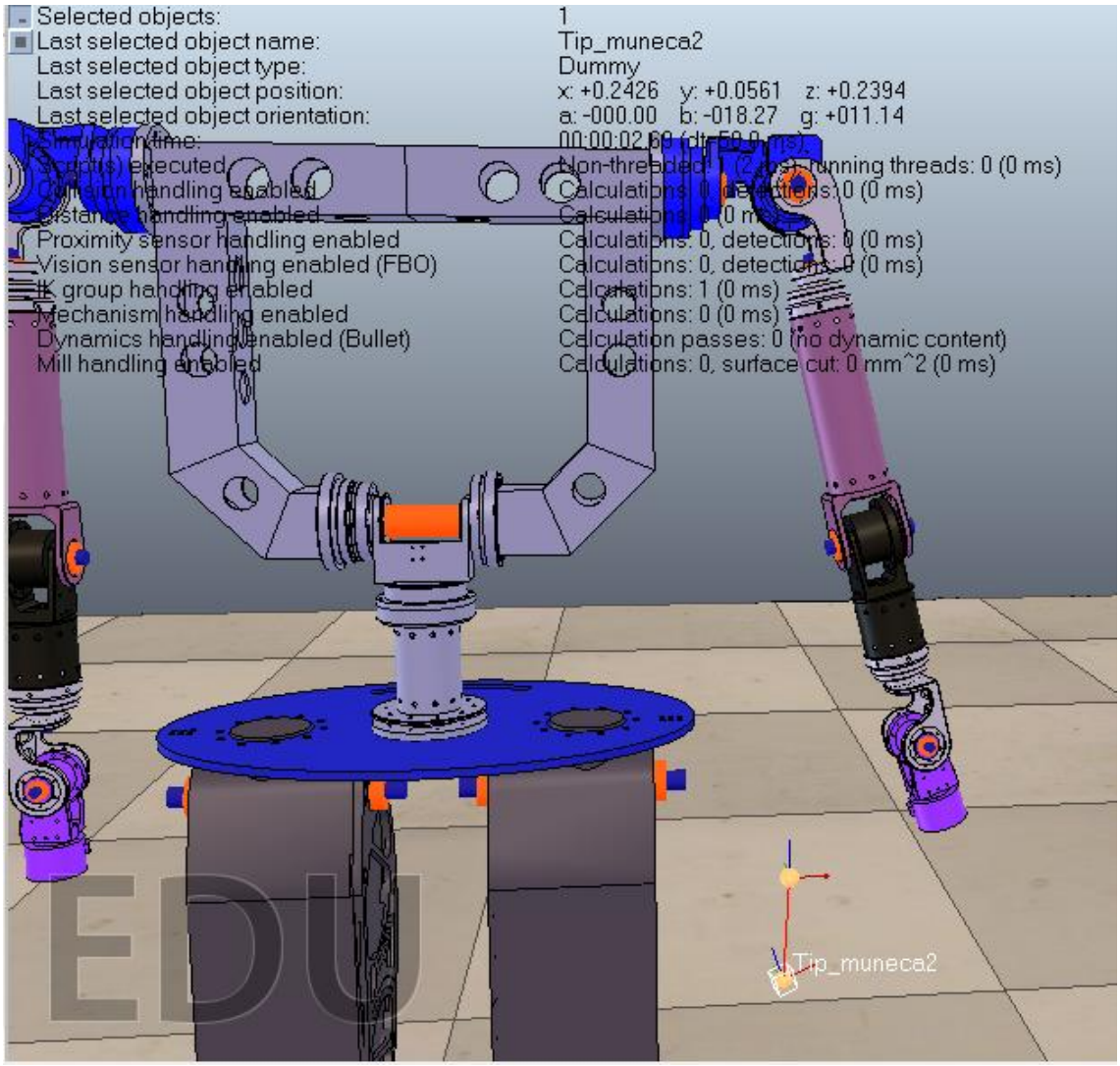

Figura 57: Simulación bajo cinemática inversa

5.8 Programación del modelo del robot

Ubicamos el robot serial creado anteriormente, para construir la trayectoria que va seguir el robot utilizamos el módulo de generación de trayectorias. Para ello seleccionamos la opción “Path” del menú “Add” ubicado en el menú principal de la aplicación y de los dos tipos posibles de caminos que se pueden crear seleccionamos la opción “cyrcle type” y podemos apreciar una trayectoria circular en el entorno de simulación. Podemos modificar los parámetros principales de la trayectoria, posicionándonos en el icono “Path” ubicado en la ventana de jerarquía, se activa la caja de dialogo “Path properties” y podemos apreciar las propiedades que se pueden modificar de una trayectoria tales como posición y velocidad de la trayectoria o seleccionar el algoritmo que interpola cada uno de los puntos de la misma.

Asimismo utilizando la herramienta “Position/Orientation” podemos modificar la posición y orientación de cada uno de los puntos de la trayectoria. De la opción “Edit modes” del submenú “Edit” seleccionamos la opción “Enter Path Edit Mode”, de la lista de puntos seleccionamos los que no vamos a utilizar y los eliminamos, nos posicionamos en cada uno ellos y modificamos la posición y orientación con la herramienta “Position/Orientation” según como deseemos.

Una vez tengamos la trayectoria deseada la guardamos dejando el modo de edición que aplica los cambios realizados (ver figura 58).

Relacionamos el elemento base con el elemento efector del robot. Creamos un elemento Dummy que va representar el punto en el espacio en donde el robot se ubicará.

Para que el robot realice la tarea deseada tenemos que programarlo para definir en qué momento moverse, para tal fin tenemos que escribir un programa utilizando el lenguaje de programación Lua propio de la herramienta V-Rep.

Después de haber escrito y asociado los programas a cada uno de los elementos necesarios en la escena todo está listo para ejecutar la simulación. Seleccionamos la opción “Play” para visualizar la simulación y sus resultados.

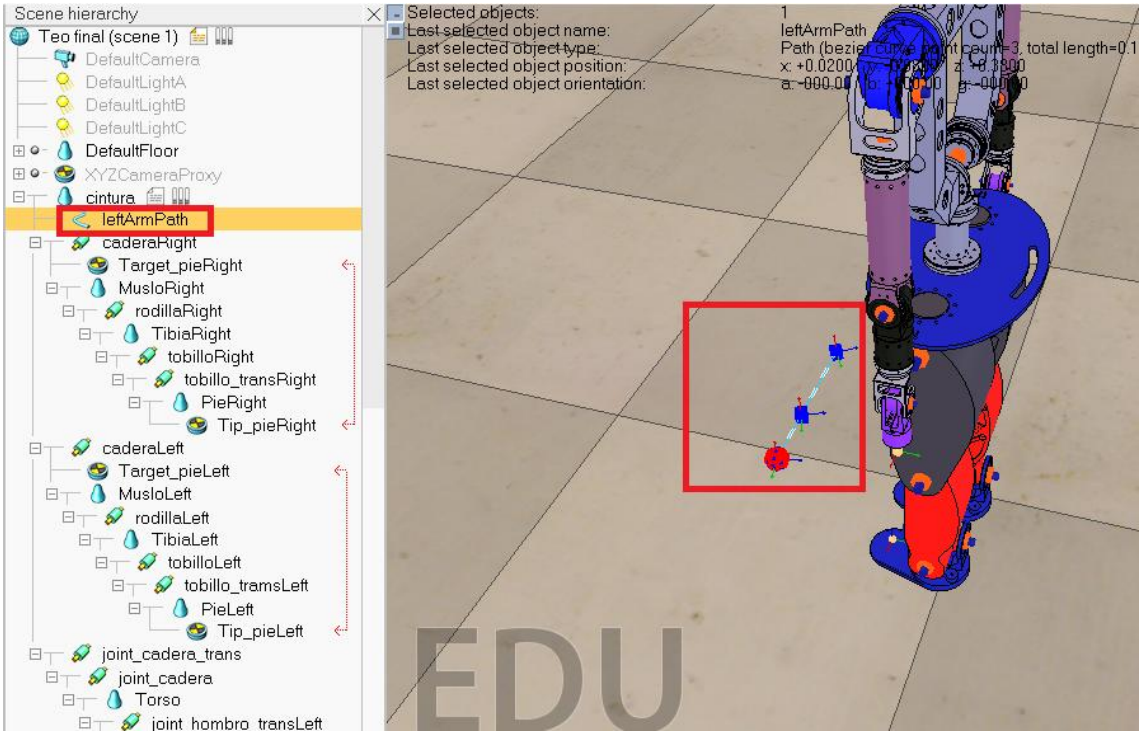

Figura 58: El Path y su trayectoria

5.8.1 LUA

Lua es un lenguaje de programación extensible diseñado para una programación procedimental general con utilidades para la descripción de datos. También ofrece un buen soporte para la programación orientada a objetos, programación funcional y programación orientada a datos. Se pretende que Lua sea usado como un lenguaje de script potente y ligero para cualquier programa que lo necesite.

Siendo un lenguaje de extensión, Lua no tiene noción de programa principal (main): sólo funciona embebido en un cliente anfitrión, denominado programa contenedor o simplemente anfitrión (host).

La semántica de Lua puede ser extendida y modificada redefiniendo funciones de las estructuras de datos utilizando meta tablas, casi como en Perl. Lua ofrece soporte para funciones de orden superior, recolector de basura. Combinando todo lo anterior, es posible utilizar Lua en programación orientada a objetos.

Los programas en Lua no son interpretados directamente, sino compilados a código bytecode, que es ejecutado en la máquina virtual de Lua. El proceso de compilación es normalmente transparente al usuario y se realiza en tiempo de ejecución, pero puede hacerse con anticipación para aumentar el rendimiento y reducir el uso de la memoria al prescindir del compilador.

La distribución de Lua incluye un programa anfitrión de muestra denominado lua, que usa la biblioteca de Lua para ofrecer un intérprete de Lua completo e independiente. [18]

En la siguiente imagen (ver figura 59) vemos el programa que hemos escrito para conseguir que nuestro el brazo de nuestro robot sigua el path creado y pueda hacer una trayectoria descrita.

```
if (simGetScriptExecutionCount()==0) then

    cintura=simGetObjectHandle("cintura")
    lArm=simGetObjectHandle("target_munecaleft")
    lPath=simGetObjectHandle("leftArmPath")
    lPathLength=simGetPathLength(lPath)
    dist=0

    nominalVelocity=0.2
    leftArmJoints={simGetObjectHandle("joint_hombro_transleft"),
    simGetObjectHandle("joint_hombroleft"),simGetObjectHandle("joint_brazoleft"),
    simGetObjectHandle("joint_codoleft"), simGetObjectHandle("joint_antebrazoleft"), simGetObjectHandle("joint_munecaleft"


end

    simHandleChildScript(sim_handle_all_except_explicit)


    -- Get the desired position and orientation of each foot from the paths (you can also use a table of values for that):
    t=simGetSimulationTimeStep()*nominalVelocity
    dist=dist+t
    lPos=simGetPositionOnPath(lPath,dist/lPathLength)
    lOr=simGetOrientationOnPath(lPath,dist/lPathLength)

    cinturaM=simGetObjectMatrix(cintura,-1)
    cinturaMInverse=simGetInvertedMatrix(cinturaM)

    m=simMultiplyMatrices(cinturaMInverse,simBuildMatrix(lPos,lOr))

    m=simMultiplyMatrices(cinturaM,m)
    lPos={m[4],m[8],m[12]}
    lOr=simGetEulerAnglesFromMatrix(m)

    simSetObjectPosition(lArm,-1,lPos)
    simSetObjectOrientation(lArm,-1,lOr)
```

Figura 59: Programación de Teo

Una vez compilado el programa podemos simular en V-Rep y ver como el brazo se mueve en su conjunto hasta la trayectoria marcada, añadimos una imagen del robot con el brazo izquierdo después de haber encontrado el path descrito anteriormente (ver figura 60 y 61).

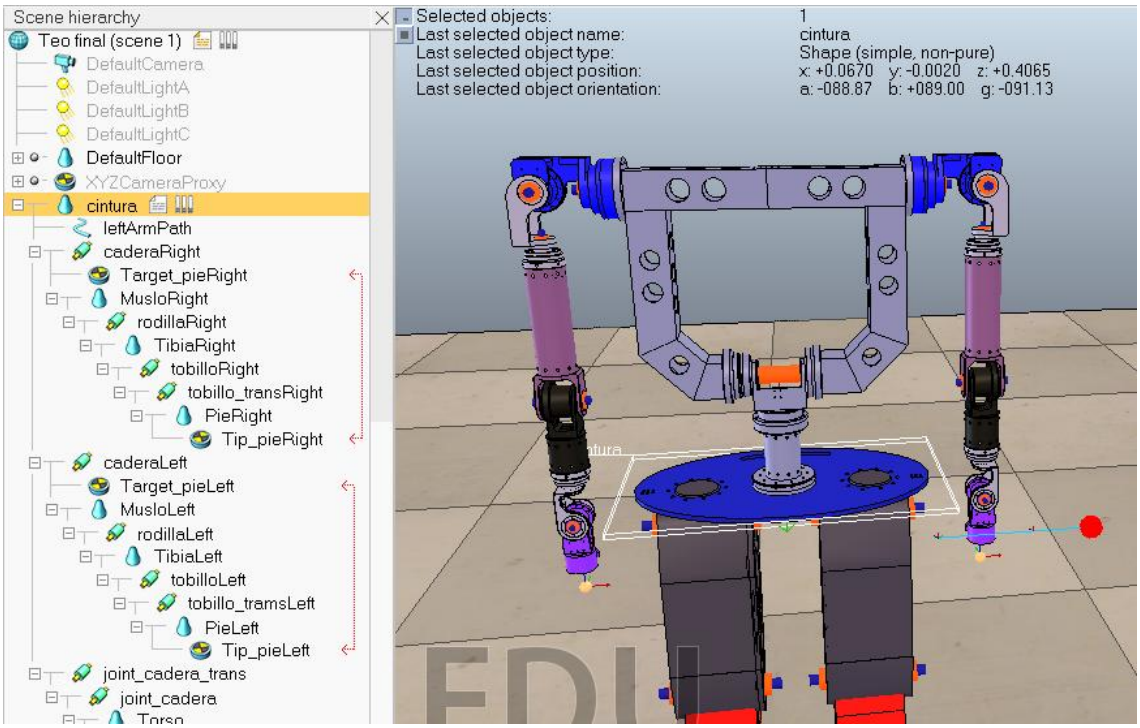

Figura 60: Teo en reposo

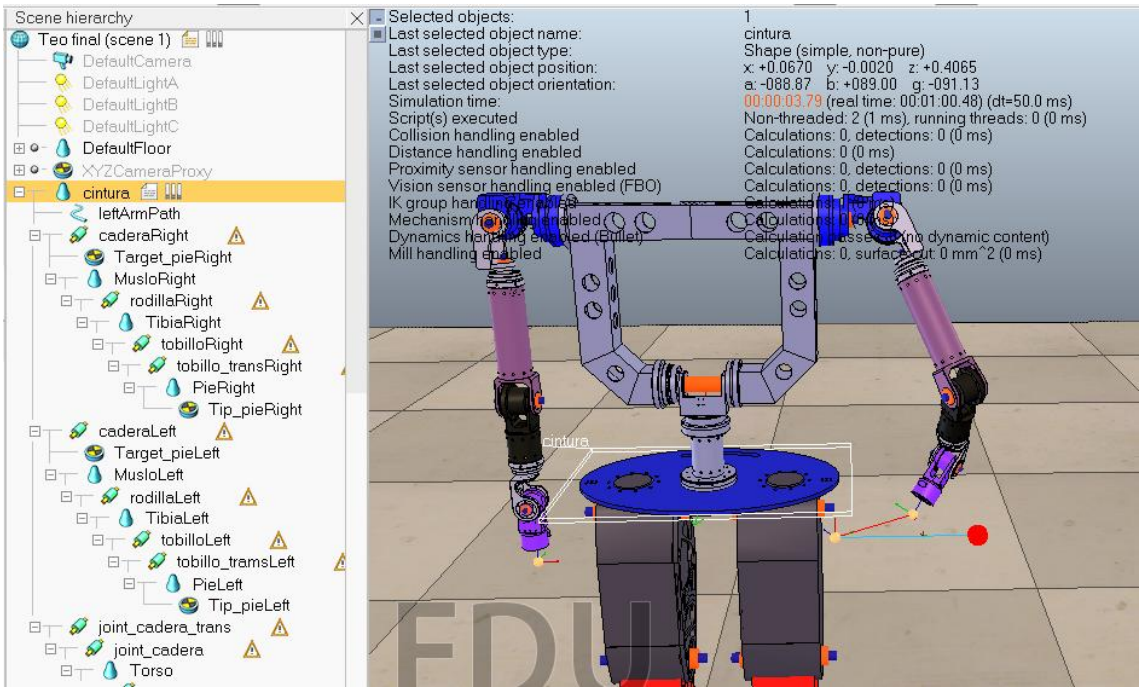

Figura 61: Teo en simulación

6 Construcción del Modelo

6.1 Robot Humanoide TEO

El robot Teo original, de donde sacamos las referencias para construir nuestro robot virtual están indicadas en la tabla 1. Estos valores fueron calculados por el departamento de robótica de la Universidad Carlos III de Leganés cuando diseñaron el robot original.

Cada pieza viene descrita por su longitud en milímetros, siendo cada "L" las piezas que constituyen el robot, por lo que hemos diseñado cada pieza del robot con estas medidas para conseguir un modelo lo mas similar posible al robot real.

	L1	L2	L3	L4	L5	L6	L7	L8	L9
[mm]	330	33,22	300	124	146	305	338	337	210

Tabla 1: Medidas de las piezas del robot humanoide TEO.

TEO es un robot mecánico que posee 28 grados de libertad (GDL). En la siguiente imagen podemos observar cómo se distribuyen los grados de libertad del robot, en la imagen solo aparecen 26 GDL porque no está incluido el cuello con sus dos grados de libertad. Se puede observar la estructura de brazos, torso, cadera y piernas a las distancias que anteriormente hemos comentado que están originalmente en el robot real (ver figura 62).

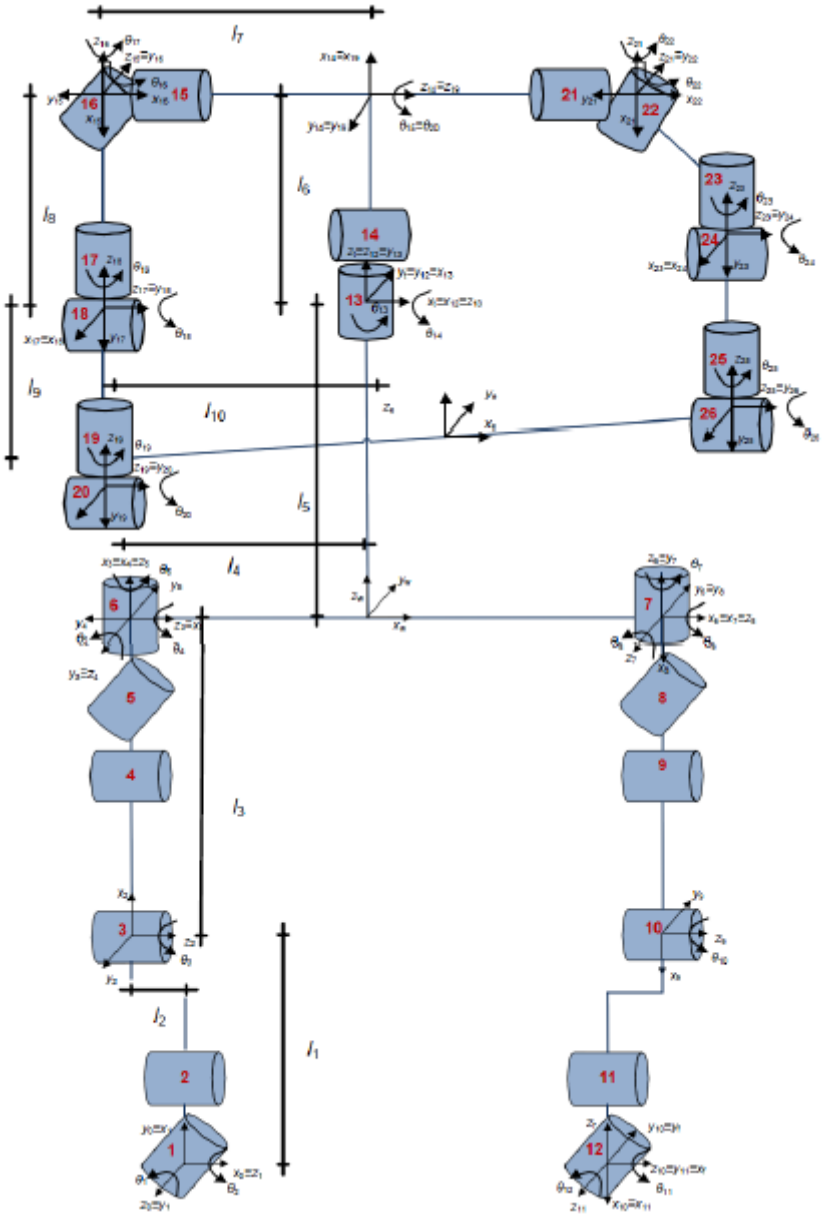

Figura 62: Distribución de los GDL del robot Teo

En la figura 36 se muestra el modelo virtual (ver figura 63), su diseño ha sido adaptado a las medidas y forma originaria por el robot Teo, usando la herramienta Solidworks para diseñar y ajustar todos los eslabones del robot a su diseño real, y en el simulador V-REP podemos ver el ensamblaje final de todos sus eslabones, dando como resultado un robot humanoide de estructura, tamaño y dimensiones similar al robot real.

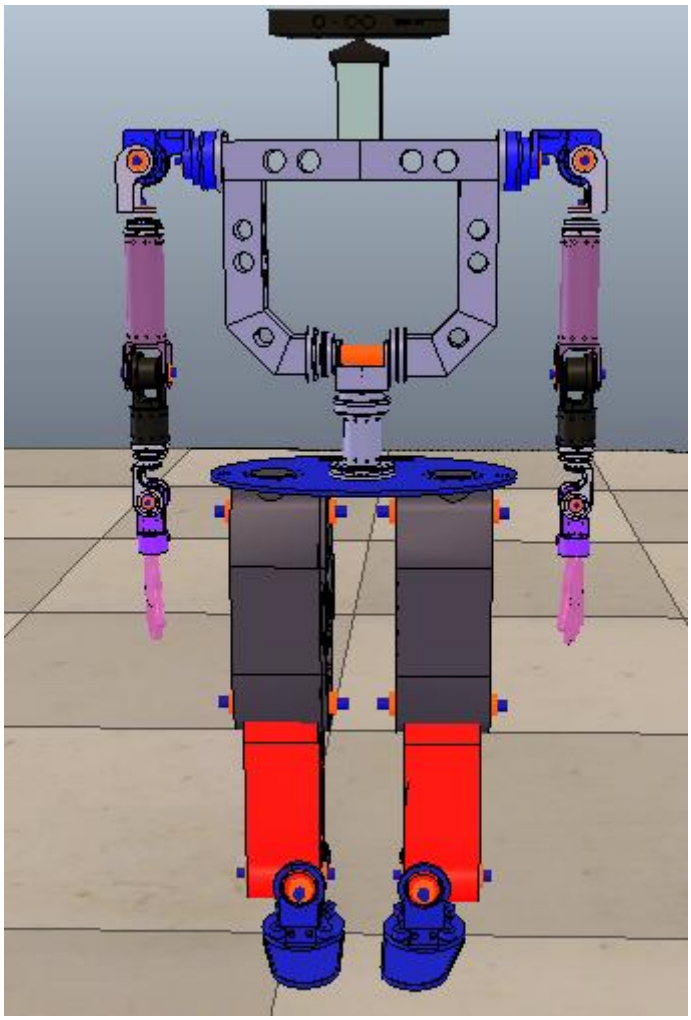

Figura 63: Modelo del Robot Teo en V-Rep

En la figura 64 se muestra el robot TEO real en el laboratorio de la universidad Carlos III de Leganés.

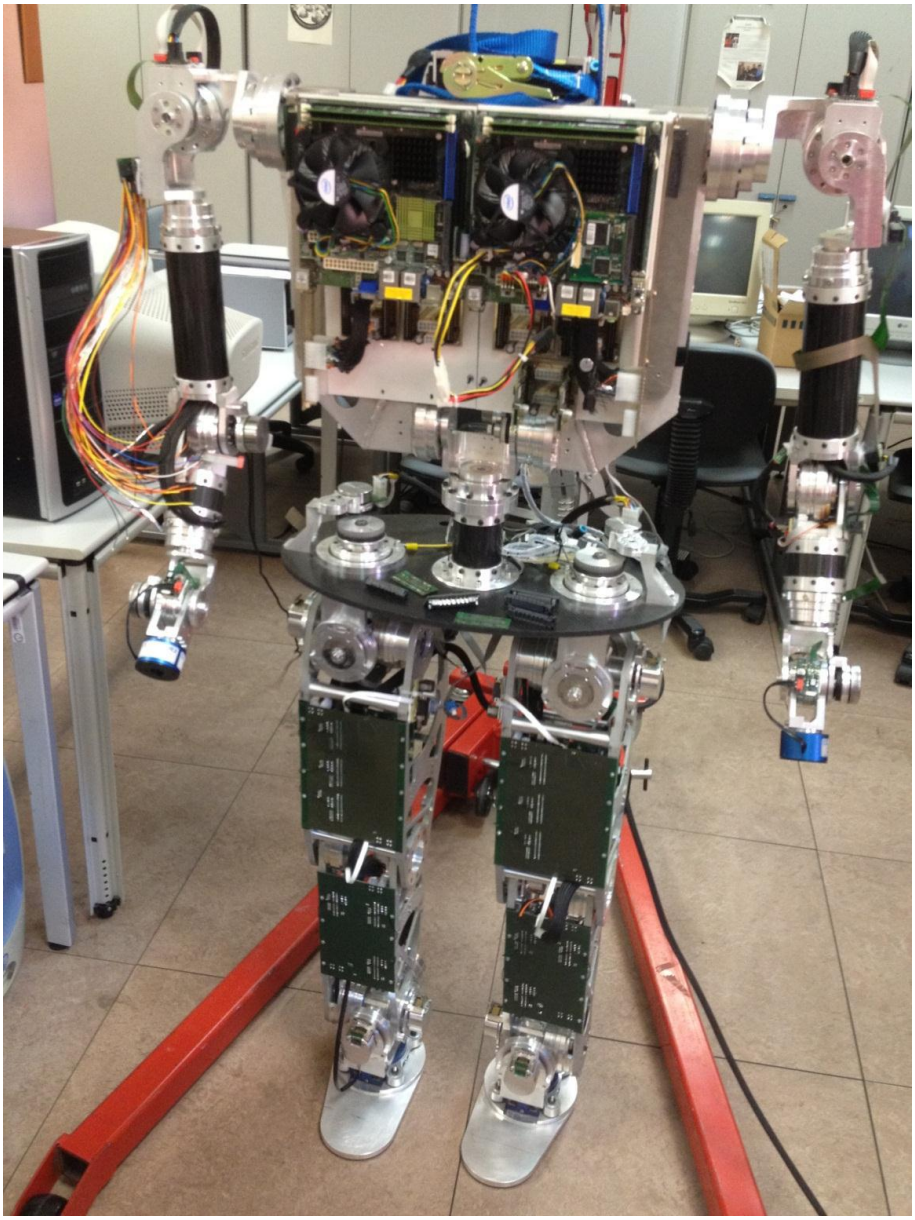

Figura 64: Robot Teo real

6.2 Grados de libertad del Robot TEO

6.2.1 Grados de libertad de la pierna

De los 28 grados de libertad, en cada pierna podemos encontrar 6 GDL. Cada pierna está formada por tres eslabones que conforman esos grados de libertad, entre ellos está la cadera, la rodilla y el tobillo de cada pierna. En la figura 65 se muestra una imagen de la parte inferior del robot Teo, se puede observar que las piernas del robot que constan de 6 grados de libertad cada una, el total de la parte inferior del robot está formado por 12 GDL con lo que se consigue que el robot pueda desplazarse en línea recta e imitar la forma de caminar humana.

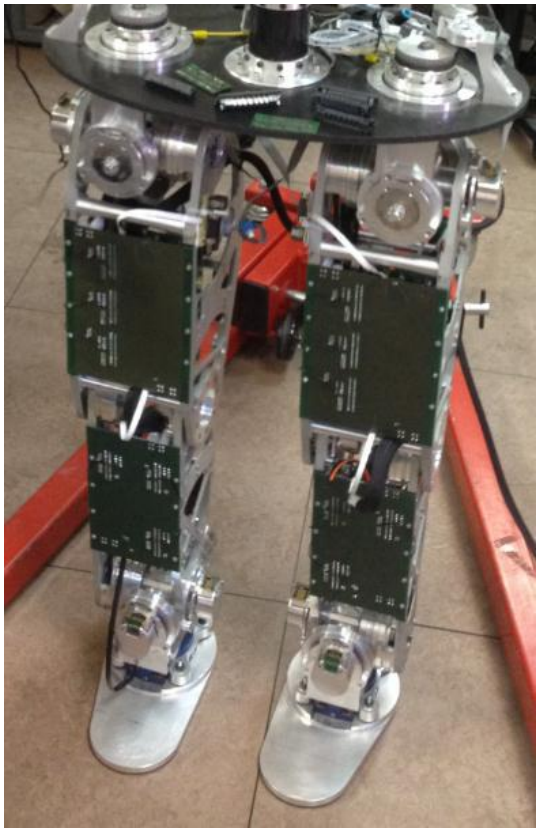

Figura 65: Parte inferior del robot Teo

6.2.1.1 Cadera

En la cadera vamos a encontrar el mayor número de grados de libertad. Está formada por 3 GDL y todos son rotatorios, sobre el eje X, Y e Z, tal como se muestra en la figura 66.

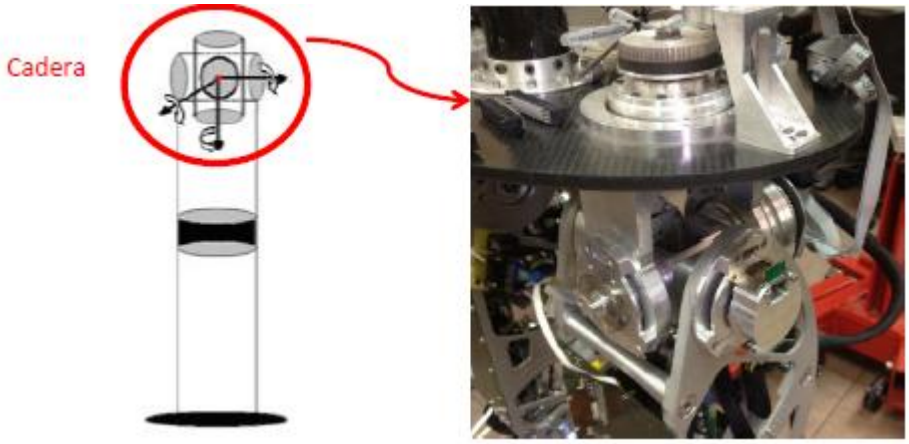

Figura 66: GDL de la cadera

La articulación que rota con respecto al eje X se utiliza para que la pierna se pueda mover de delante a atrás, la articulación sobre el eje Y se necesita para que la pierna pueda rotar sobre si misma y la articulación sobre el eje Z permite a la pierna moverse de lado completando en conjunto el movimiento necesario para desplazarse.

6.2.1.2 Rodilla

La rodilla solo posee un grado de libertad, es la parte mecánica más fácil de implementar, todo su movimiento rota sobre un eje como podemos ver en la figura 67.

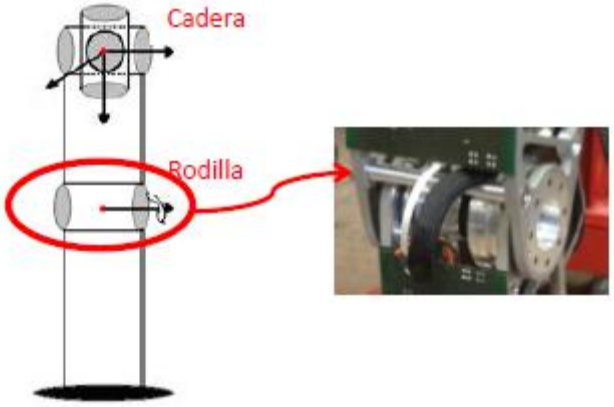

Figura 67: GDL de la rodilla

6.2.1.3 Tobillo

El tobillo está formado por 2 GDL, estos dan movimiento en el eje X y Z (ver la figura 68), el grado de libertad del eje X es más importante que el Z, puesto que este grado de libertad es el que permite balancear el peso del robot hacia el pie de apoyo, para que el centro de masa del robot se encuentre en todo momento dentro del área de soporte del pie sobre el terreno (balanceo estático).

El grado de libertad en el eje Z es útil para contrarrestar el peso, ya sea hacia delante o hacia atrás cuando uno o los dos pies se encuentren sobre el suelo.

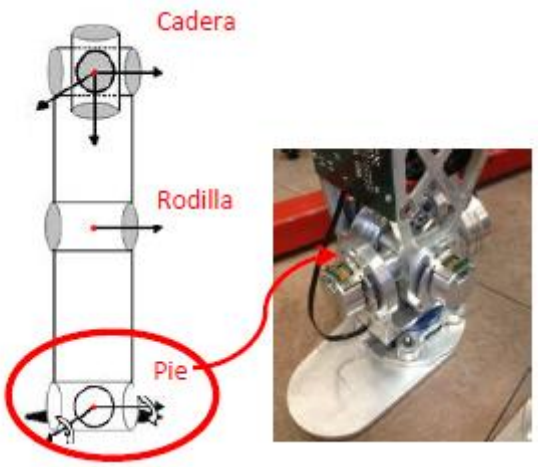

Figura 68: GDL del tobillo

6.2.2 Grados de libertad del brazo

Los brazos o extremidades superiores están formados por 6 GDL cada uno, estos están segmentados entre el hombro, el codo y la muñeca. Cada brazo puede moverse de manera conjunta o independiente en función de la necesidad. En la siguiente foto (figura 69) se muestran el brazo izquierdo del robot TEO.

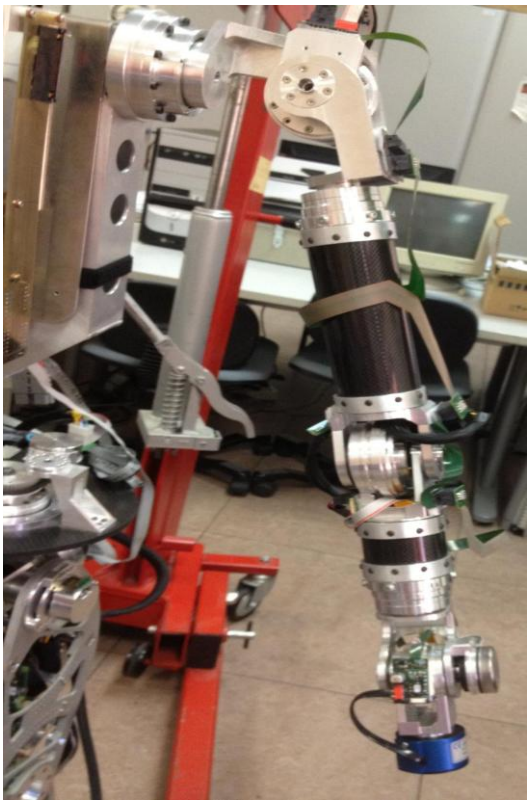

Figura 69: Brazo del robot Teo

6.2.2.1 Hombro

En el hombro encontramos dos grados de libertad, estos ayudan a mover el brazo en todos los sentido, hacia delante hacia atrás y levantarlo de lado hacia afuera. En la figura 70 podemos ver una imagen del hombro.

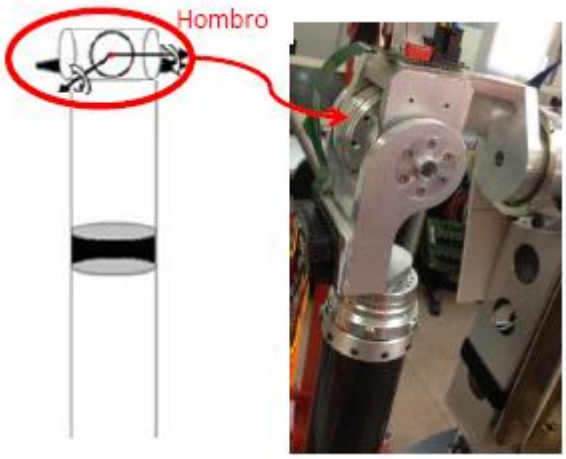

Figura 70: GDL del hombro

6.2.2.2 Codo

El codo está formado por dos grado de libertad, uno específico del codo que ayuda a girar en el eje X para mover el antebrazo y otro sobre el eje Y para ayudar a que el brazo roto sobre si mismo tal como se muestra en la figura 71.

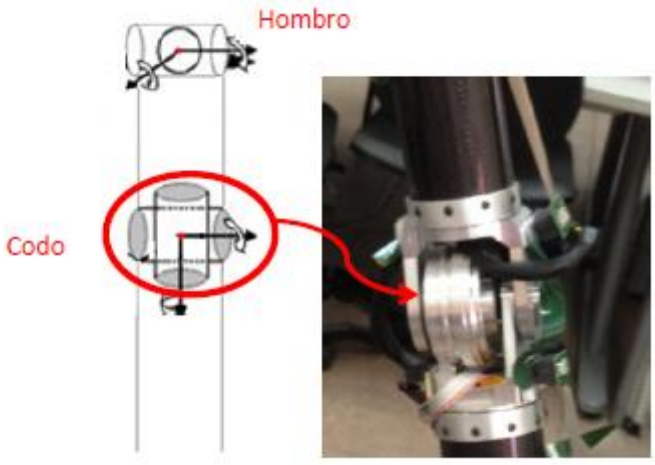

Figura 71: GDL del codo

6.2.2.3 Muñeca

La muñeca es similar al codo, está formada por dos grados de libertad uno en el eje X para ayudar al girar la muñeca para levantar la mano y otro en el eje Y para girar la mano y el antebrazo sobre sí mismos como se muestra en la Figura 72.

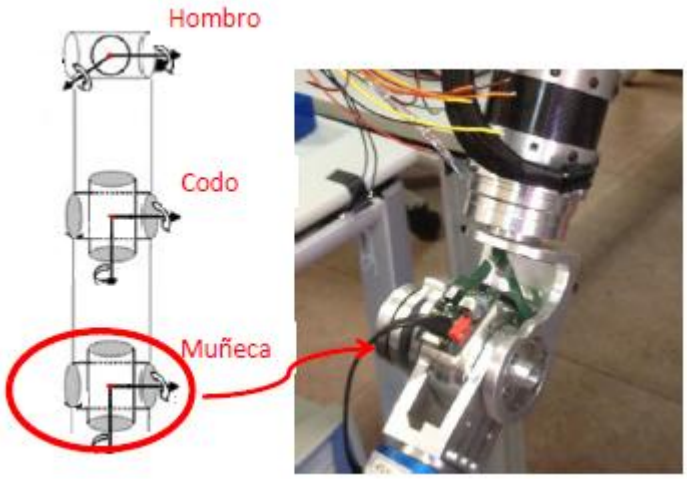

Figura 72: GDL de la muñeca

6.2.3 Grados de libertad del tronco

El tronco está formado por dos grados de libertad, uno en el plano Y que le permite el giro de la parte superior del cuerpo sobre sí mismo sin tener que mover las piernas, y otro en el plano X que le permite regular su inclinación (ver la figura 73).

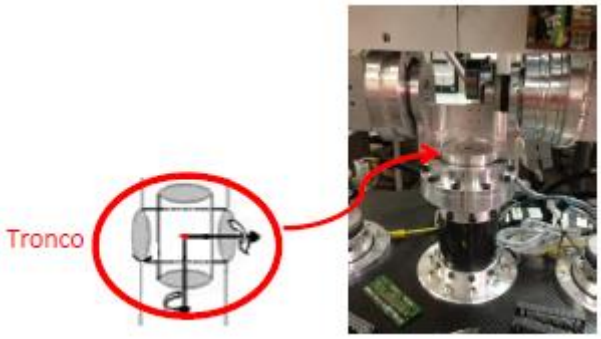

Figura 73: GDL del tronco

6.2.4 Grados de libertad del cuello

El cuello posee dos grados de libertad, uno en el plano Y que le permite el giro la cabeza sobre sí mismo de lado a lado, y otro en el plano X que le permite regular su inclinación de arriba a abajo (ver la figura 74).

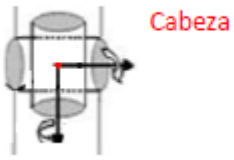

Figura 74: GDL del cuello

A continuación en la figura 75 se muestran todas las articulaciones creadas en V-REP para conseguir la simulación de movimiento del modelo TEO.

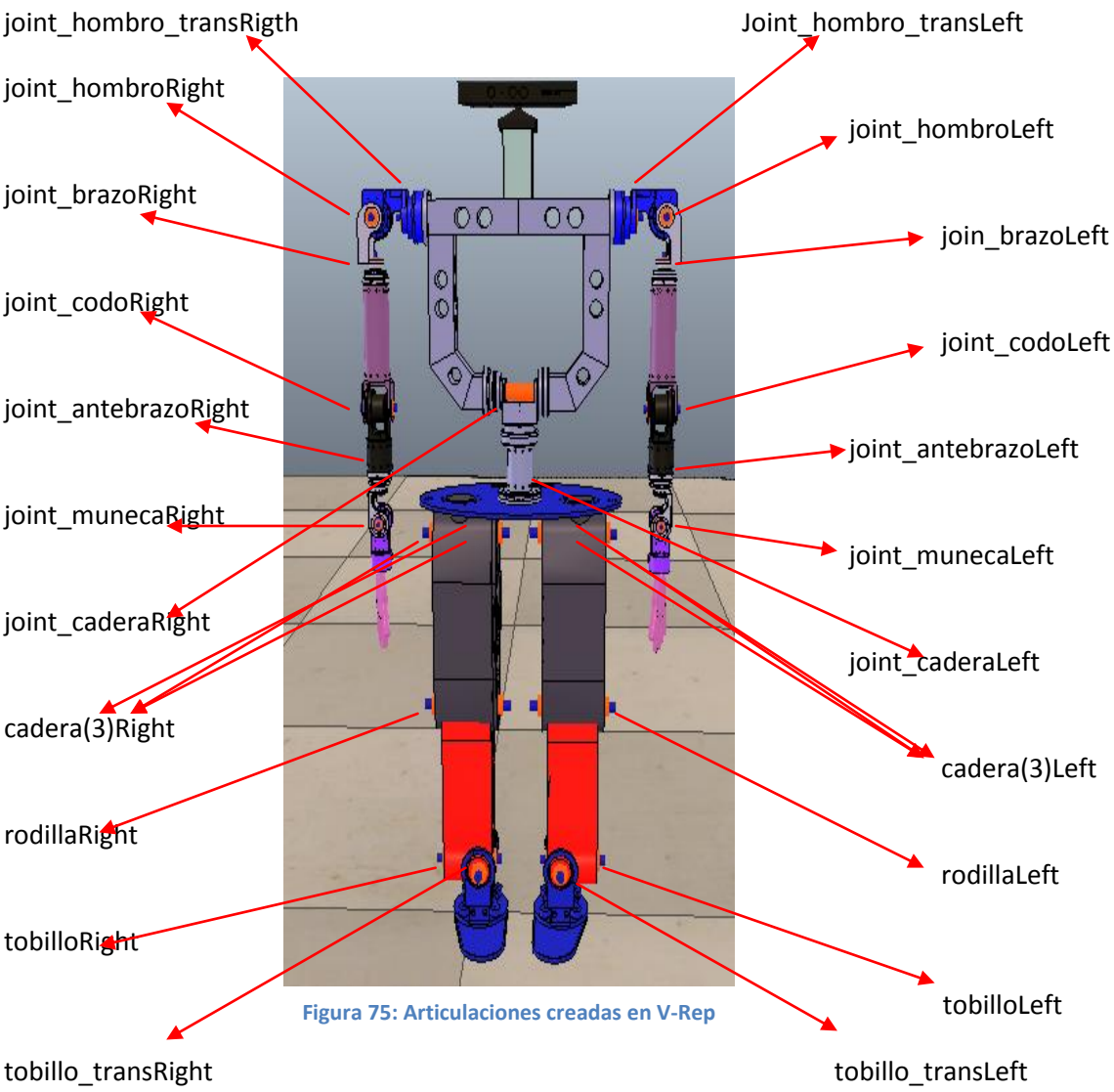

Figura 75: Articulaciones creadas en V-Rep

6.3 Eslabones del robot TEO

6.3.1 Cuerpo

TEO se divide en dos nodos base, el torso y la cadera, siendo hijos de estos están los siguiente cuatro nodos de cada articulación, brazos y piernas que conforman el cuerpo del robot. La unión de los nodos base, están ensamblados mediante las articulaciones "joint_cadera" articulación vertical que permite el movimiento del torso frente a la cadera con respecto al eje Z, y una articulación "joint_cadera_trans" que permite el movimiento con respecto al eje X.

6.3.2 Brazos

Cada brazo está dividido en 6 partes principales y estas a la vez compuestas por distintas partes, la primera parte está compuesta por el hombro formado por dos elementos que crean el hombro en su conjunto, "hombro_torso" unida al torso por una articulación "joint_hombro_trans"y "hombro_biceps" unida a la otra parte del hombro por la articulación "joint_hombro". La segunda parte del brazo es el bíceps "bíceps_codo" unido al hombro por una articulación llamada "join_brazo". La tercera parte del brazo es el codo "codo_antebrazo" unida al bíceps por la articulación "joint_codo". La cuarta parte del brazo es el "antebrazo_muneca" unida al codo por la articulación "joint_antebrazo". La quinta parte del brazo es la muñena"muneca" unida al antebrazo por la articulación "joint_muneca". Y la sexta y última parte que forma el brazo robótico de TEO es la mano "mano" unida a la muñeca y acabando así el eslabón completo de cada brazo.

6.3.3 Cadera

La cadera es un eslabón único con forma esférica donde se unen la parte superior del cuerpo que es el torso y la parte inferior que son la piernas, tiene una articulación unida al torso que anteriormente mencionamos "joint_cadera" y "joint_cadera_trans".

6.3.4 Piernas

Cada pierna está dividida en 3, la primera parte está compuesta por el "muslo", unida a la cadera por una articulación "cadera". La segunda parte de la pierna es la "tibia" unido al muslo por una articulación llamada "rodilla". La tercera parte de la pierna es el "pie" unido a la tibia por dos articulaciones para conseguir todos los movimientos del pie llamados "tobillo" y "tobillo_trans". Con estas tres partes formamos las piernas ambas simétricas en forma y estructura.

6.3.5 Cabeza

La cabeza es la unión de dos elementos un prisma unido al torso del cuerpo que hace de cuello y a este unida la cabeza que tiene forma de una Kinet que es lo que el robot TEO tiene por cabeza para visualizar el entorno.

En la figura 76 se muestra el modelo del robot donde se señalan los eslabones que terminan de completar el modelo.

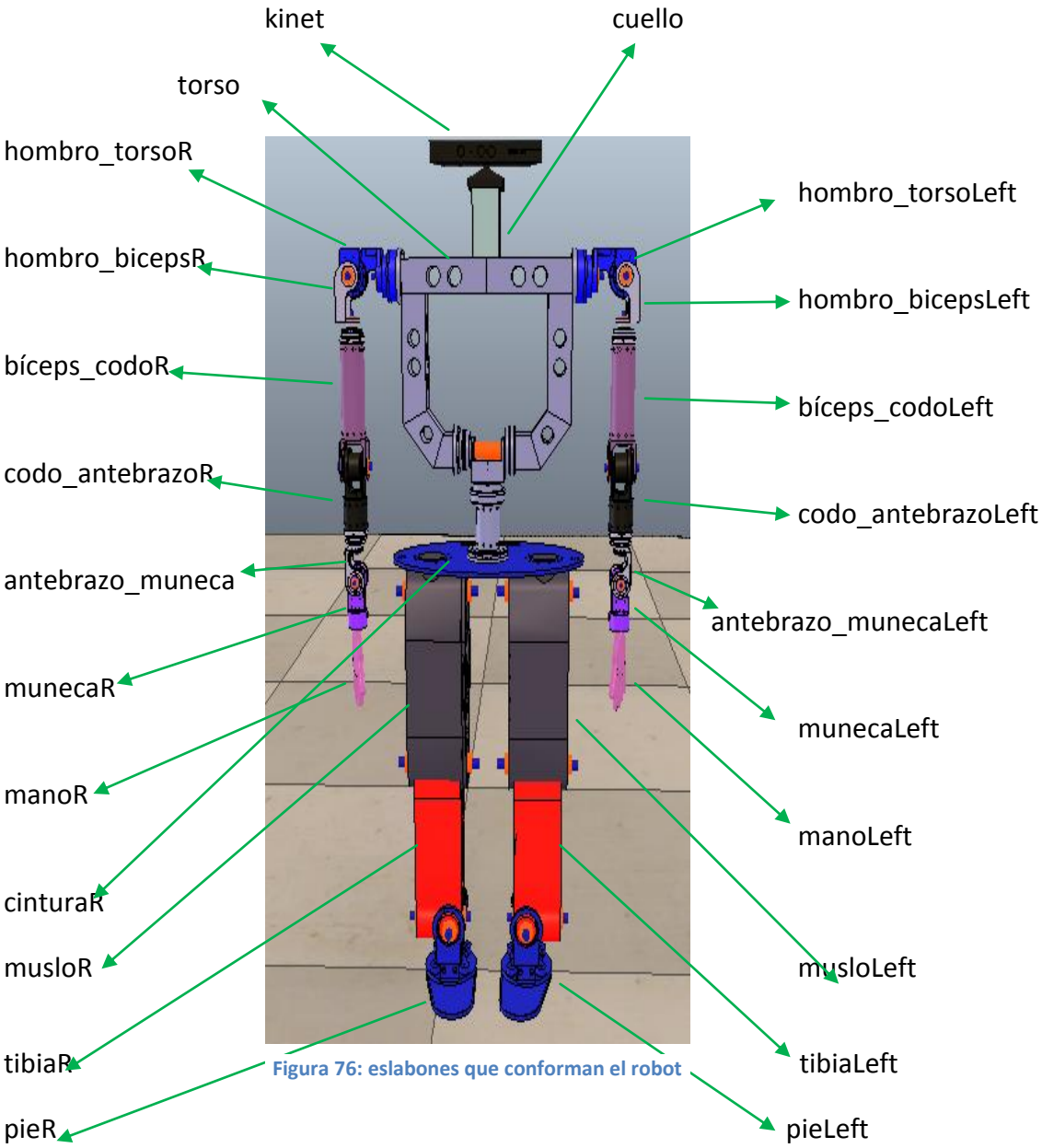

Figura 76: eslabones que conforman el robot

7 Conclusiones

Este proyecto de robot humanoide, se ha creado con la base de poder modelar y simular virtualmente a Teo, un robot real de la universidad Carlos III de Leganés, para posteriormente poder aplicar la programación y desarrollo creado en el simulador en un entrono real. El objetivo del proyecto ha estado orientado en el modelado de Teo en Solidworks para crear pieza a pieza el sólido que importaremos al simulador V-Rep. Una vez hecha la importación hay que ensamblar el modelo para conseguir la mayor exactitud en medidas y estructura al modelo real, este modelo se ha desarrollado siguiendo las características cinemáticas del robot humanoide TEO.

Para ello ha sido necesario familiarizarse con el entorno de trabajo por medio de un estudio detallado del simulador V-REP y así poder comprender todo su funcionamiento. Como apoyo en V-Rep. tenemos el manual de usuario donde podemos conseguir ayuda. El programa también tiene varios ejemplos que se fueron consultando y realizando progresivamente añadiendo características al robot creado.

Para construir dicho modelo, hay que conocer sus nodos y el funcionamiento del robot TEO, para ello, diseñamos las estructuras jerárquica de los nodos del robot TEO, tomando como base la cintura para la parte inferior del cuerpo que las extremidades inferiores, y el torso para la parte superior. Debido al movimiento de todas las articulaciones de robot hemos usado juntas rotacionales.

Una vez desarrollado el modelo, mediante la herramienta de diseño Solidworks, y ha sido importado en V_REP eslabón a eslabón para luego poder ensamblarlo, el objetivo era conseguir que los eslabones se movieran en su conjunto consiguiendo un ensamblaje perfecto de las articulaciones y los eslabones, para comprobar estos movimientos hemos usados dos métodos de simulación. El primer método es la cinemática inversa mediante un “tip” y un “target” para guiarlo en la posición donde alojemos el “target” y el segundo método ha sido crear un “path” asociándolo a una trayectoria para que el robot bajo la programación en Lua del movimiento de una parte del robot siguiera la trayectoria marcada por el path.

Finalmente se ha grabado varios videos de los movimientos que hemos simulado con el modelo de robot TEO en V-REP.

En conclusión, se pueden considerar los objetivos del trabajo fin de grado propuesto como conseguidos.

8 Trabajos futuros

Los siguientes pasos serían incorporar al modelo creado avances tecnológicos como cámaras y sensores para conseguir que el robot pueda realizar tareas más complejas y específicas del ser humano.

Dentro de esas tareas, lo principal es conseguir que el robot pueda caminar y pueda realizar tareas más complejas como coger objetos y analizarlos, para ello es necesario añadir al robot una serie de sensores que le permitan realizan con efectividad esas tareas como por ejemplo:

Añadir un sensor inercial (IMU), para medir la aceleración y la velocidad angular de los movimientos del robot.

Un sensor FUERZA/PAR: Los robots requieren de este tipo de sensores para obtener información durante los movimientos y poder calcular y ajustar la posición de los actuadores encargados del equilibrio.

En la parte de la cabeza, en mi diseño e incorporado una imagen de la Kinect, para que en trabajos futuros esa Kinect actúe como cámara de alta resolución y pueda analizar visualmente el entorno. Kinect cuenta con una cámara de video a tres colores y sensor de profundidad que es capaz de reconocer la cara humana y ver el espacio en tres dimensiones bajo cualquiera condición de iluminación.

Importante es incluir a cada eslabón una masa para conseguir que el modelo se desplace y simule la caminata de un ser humano.

Para simular un entorno real, sería ideal crear entornos de simulación de la vida real como por ejemplo el salón de una vivienda, lugares públicos o espacios donde pueda moverse haciendo deporte.

9 Bibliografía

9.1 Páginas Web:

[1] Robótica

http://es.wikipedia.org/wiki/Robots

[2] Clasificación de los robots

http://usuarios.lycos.es/sparta/experiences12.html

[3] Nao Robot

http://www.aldebaran.com/en/more-about

[4] New Asimo

http://asimo.honda.com/news/

[5] HRP-4C

http://www.xatakaciencia.com/robotica/hrp-4c-miim-el-robot-con-apariencia-humana

[6] Partner Robot

http://es.wikipedia.org/wiki/Toyota_Partner_Robot

[7] QRIO

http://www.roboticspot.com/robots.php?id=2

[8] Solidworks

www.solidworks.com

[9] SimRobot

http://www.informatik.uni-bremen.de/simrobot/index_e.htm

[10] Gazebo

http://gazebosim.org/

[11] OpenHRP3

http://www.openrtp.jp/openhrp3/en/about.html

[12] Marilou Robotics Studio

http://www.anykode.com/index.php

[13] Microsoft Robotics Developer Studio 4

http://es.wikipedia.org/wiki/Microsoft_Robotics_Studio

[14] OpenRAVE

http://en.wikipedia.org/wiki/OpenRAVE

[15] Webots

http://www.cyberbotics.com/

[16] V-REP User Guide (release 3.0.1) copyright (c) 2011 Coppelia Robotics. All rights reserved.

www.coppeliarobotics.com

[17] V-REP Reference Manual (release 3.0.1) copyright (c) 2011 Coppelia Robotics. All rights reserved.

www.coppeliarobotics.com

[18] Lua

http://www.lua.org/manual/5.1/es/manual.html